Feb./Mar./Aprile 1990, Anno V n. 2

La rivista per utenti di C-64/128 ed Amiga

COMMODORE GAZETTE

Computergrafica:

GRAFICA STRUTTURATA E CAD CON L'AMIGA

Dai nostri inviati:

- IL WORLD OF COMMODORE DI TORONTO
- LA CONFERENZA DEI PROGRAMMATORI DI PARIGI

Computermusica:

IL COMPLETE SOUNDSCAPE MIDI MUSIC SYSTEM

Listati per C-64/128:

- FAST-40
- DUO
- NICK 128

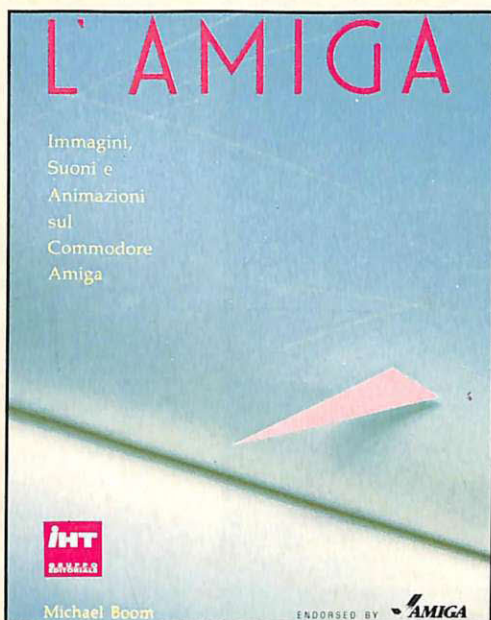
Amiga, primi passi:

- PARTIAMO DA ZERO
- L'EDITOR MEMACS

Programmare l'Amiga:

- LA LIBRERIA ARP
- UN DATABASE IN F-BASIC
- GLI OFFSET ALLE STRUTTURE DELL'AMIGABASIC

Sped. in abb. post. Gr. III/70 Lire 5.000

Un computer chiamato AMIGA

L'AMIGA

Come il Commodore Amiga ha cambiato il mondo dei computer, così il volume *L'Amiga* proietterà la vostra immaginazione lungo nuovi orizzonti di creatività. Preparatevi a ottenere stupefacenti risultati dal vostro computer: sofisticate immagini video, suoni e musica, sequenze animate da registrare su videocassetta, e molto di più. In questo volume troverete inoltre consigli utili per ottenere il meglio dall'Amiga Basic e da prodotti software eccezionali come Deluxe Paint, Deluxe Music e Deluxe Video.

416 pagine, oltre 100 illustrazioni, L. 60.000

IL MANUALE DELL'AMIGADOS

Questo volume è l'unica documentazione ufficiale realizzata dalla Commodore sul sistema operativo dei computer Amiga. Contiene tre libri (Il Manuale per l'utente, Il Manuale per il programmatore e Il Manuale di riferimento tecnico) che costituiscono nel loro complesso la guida più completa per ogni utente dell'Amiga, dal principiante che vuole usare i comandi dell'AmigaDOS (che non sono spiegati dall'opuscolo in dotazione al computer), fino al programmatore evoluto che troverà utili informazioni per programmare in C e in Assembly.

376 pagine, L. 60.000

Sommario

FOTO PATRICIA LEEDS

FEB./MAR./APRILE 1990
Anno V, N.2

ARTICOLI

RUBRICHE

COMMODORE GAZETTE
è una pubblicazione
Direttore responsabile: Massimiliano M. Lisa
Redazione: Mauro Gaffo, Dario Tonani, Nicolò Fontana Rava
Collaborazione editoriale: Fabio Rossi, Filippo Moretti, Dario Zanon, Avelino De Sabbata, Fulvio Piccioli, Alessio Palma, Alfredo Prochet, Michael Moosleitner, Stefano Franzato
Corrispondenti USA: William S. Freilich, Daniela D. Freilich
Collaborazione editoriale USA: Matthew Leeds, Ervin Bobo, Eugene P. Mortimore, Harriet Maybeck Tolly, Morton A. Kevelson. *AmigaWorld*: Bill Catchings, Mark L. Van Name, Bryan D. Catley. *Run*: Robert Cook, Bret Timmins, Mark Jordan
Segretaria di redazione: Giovanna Varia
Impaginazione e grafica: Andrea De Michelis
Fotografia: A.&D., Patricia Leeds
Disegni: Marco Piazza, Giuseppe Festino

Direzione, Redazione, Amministrazione: IHT Gruppo Editoriale S.r.l. - Via Monte Napoleone, 9 - 20121 Milano
Fotocomposizione: IHT Gruppo Editoriale S.r.l. - Divisione grafica
Fotolito: Colour Separation Trust S.r.l. - Via Melchiorre Gioia, 61 - 20124 Milano
Stampa: Nascor S.p.A. - Via Senigallia - 20161 Milano
Distribuzione per l'Italia: Messaggerie Periodici S.p.A. - V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467545
Distribuzione per l'estero: A.I.E. S.p.A. - Via Gadames, 89 - 20151 Milano - Tel. 02/3012200

Pubblicità: IHT Gruppo Editoriale S.r.l. - Via Monte Napoleone, 9 - 20121 Milano - Tel. 02/794181-799492-792612-794122 - Fax 02/784021 - Telex 334261 IHT I - Agente pubblicitario: Aldo Pagano Pagano

Abbonamenti: IHT Gruppo Editoriale Servizio Abbonati - Via Monte Napoleone, 9 - 20121 Milano. Linea per registrazione abbonamenti: 02/794181 799492-792612-794122
Costo abbonamenti: Italia 12 numeri L. 96.000 - 24 numeri L. 192.000 - 36 numeri L. 288.000
Estero: Europa L. 150.000 (10 numeri) Americhe, Asia... L. 200.000 (10 numeri). I versamenti devono essere indirizzati a: IHT Gruppo Editoriale S.r.l. - Via Monte Napoleone, 9 - 20121 Milano mediante emissione di assegno bancario o vaglia postale

Arretrati: Ogni numero arretrato: L. 16.000 (sped. compresa)

Autorizzazione alla pubblicazione: Tribunale di Milano n. 623 del 21/12/85. Periodico mensile. Sped. in abb. post. gr. III/70. ISSN: 0394-6991
La IHT Gruppo Editoriale è iscritta nel Registro Nazionale della Stampa al n. 2148 vol. 22 foglio 377 in data 5/6/1987

Associato alla U.S.P.I. (Unione Stampa Periodica Italiana)

NOTE EDITORIALI

L'Amiga 3000

Ormai è pronto! È solo questione di poco... ci sarà una presentazione in grande stile negli Stati Uniti, a New York, si potrà forse occhieggiarlo al CeBIT di Hannover e per il mese di giugno dovrebbe essere disponibile nei negozi. Che cosa? Ma l'Amiga 3000 naturalmente.

In un mondo in continua evoluzione come quello dell'informatica la Commodore ha pensato bene di affiancare ai modelli 500 e 2000 un grande fratello dalle caratteristiche alla super avanguardia per il mondo dei minicomputer. Microprocessore Motorola 68030 a 32 bit, la famosa risoluzione oltre i 1000 pixel rigorosamente senza flickering, multioperatività (lo Unix è di serie), sistema operativo 1.4 e infine un design elegante, a metà tra un Amiga 2000 e un 1000. Il prezzo dovrebbe superare la barriera dei 5 milioni di lire.

Innanzitutto vogliamo rassicurarvi: non si tratta della macchina nata per far morire 500 e 2000. Si tratta di un prodotto dalle prestazioni decisamente elevate e diretto soprattutto all'utenza professionale. Per far funzionare il vostro simulatore di volo preferito un 68030 non è necessario...

In compenso si tratta di una macchina che ha tutte le carte in regola per allargare la presenza dell'Amiga nel mondo professionale. Il che è sicuramente molto positivo: più l'Amiga, in tutti i suoi modelli, si diffonde, più software e hardware aggiuntivo si produce. E l'equazione che ne consegue è: più utenti = più software = più utenti felici del loro computer.

Il grande fratello della comunità Amiga non sembra avere quindi nulla di orwelliano... W l'Amiga.

L'appuntamento è quindi con il prossimo numero di *Commodore Gazette* (o al massimo con quello successivo) sul quale dovrebbe apparire la prova completa e ricca di particolari del nuovo Amiga 3000. Questo ovviamente salvo un imprevisto slittamento nell'uscita della macchina, cosa sempre possibile nel ribollente mondo dei computer. Rimanete sintonizzati.

Novità sulla rivista

Con questo numero della rivista abbiamo iniziato l'introduzione di una serie di novità che dovrebbero raccogliere il favore di tutti i nostri lettori.

Prima di tutto abbiamo considerato lo spazio. Abbiamo rilevato che nelle consuete 128 pagine del nostro periodico molto spazio prezioso veniva sprecato. Due pagine di indice, una di collaboratori, un'altra di editoriale, una di indice degli inserzionisti... Troppe pagine occupate con scarsa utilità per i lettori. Reimpostando la grafica di alcune parti di *Commodore Gazette* abbiamo quindi cercato di liberare nuovi spazi in modo da poter offrire nello stesso numero di pagine molti più argomenti.

Sempre allo scopo di offrire di più, abbiamo anche avviato una nuova serie di articoli impaginati in corpo minore che in una o due pagine offrono condensate una serie d'informazioni utili.

Inoltre, grazie a un accordo con un'importante Casa editrice straniera, abbiamo da questo numero anche la possibilità di pubblicare articoli tratti dalle riviste statunitensi *AmigaWorld* e *Run*. Mantenendoci comunque sempre decisamente indipendenti, Vi proporremo in ogni numero qualche articolo tratto da queste testate.

I listati per Commodore 64 e 128 hanno subito un'importante innovazione: niente più stampati da fotografare e da trasferire sulla pellicola necessaria per la realizzazione delle lastre con le quali viene poi stampata la rivista in rotativa (e quindi scadimento di leggibilità); niente più codici grafici di difficile interpretazione a causa della riduzione del formato dal modulo continuo. Da questo numero i listati vengono impaginati, come quelli dell'Amiga, attraverso il nostro sistema di fotocomposizione e i simboli grafici sono stati sostituiti da pratiche codifiche presentate fra parentesi graffe.

E si tratta solo dell'inizio. Molte altre novità seguiranno nei prossimi numeri. Buona lettura.

M.L.

AMATE LA COMPUTERGRAFICA, LE ANIMAZIONI, LA REGIA...?
...ALLORA QUESTO ANNUNCIO È PER VOI!

GARA PER LA CREAZIONE DI UNA

VIDEOSIGLA CON L'AMIGA

La IHT Gruppo Editoriale indice una gara per la realizzazione tramite l'Amiga di una schermata di presentazione che verrà utilizzata come sigla di apertura di filmati registrati su videocassetta.

Requisiti. Dev'essere riprodotto il seguente testo:

IHT Video

Preferibilmente (ma non è essenziale) la scritta IHT Video dev'essere compresa all'interno di un quadrato, come il marchio riprodotto qui di seguito senza la scritta GRUPPO EDITORIALE (in pratica bisogna sostituire a Gruppo Editoriale la parola Video):

Può essere sia una schermata statica, sia animata della durata non superiore a 15 secondi. Non devono assolutamente comparire altre scritte (come per esempio il nome dell'autore). Sfondo e animazioni sono a completa discrezione dell'autore. Per la sua realizzazione dev'essere utilizzato necessariamente l'Amiga con programmi a scelta dell'autore (come per esempio Deluxe Paint, Deluxe Video, Videotitler, Videoscape 3D, Animagic, Sculpt Animate 3D e 4D, Broadcast Titler, Photon Paint...). Può essere incluso anche un commento sonoro, ma non è né richiesto né indispensabile.

☆

Formato. Il prodotto finito dev'essere inviato su videocassetta VHS oppure registrato su un disco dell'Amiga possibilmente con partenza in autoboot. Nel caso la sigla di presentazione sia registrata su un disco Amiga e non parta in autoboot devono essere incluse chiare istruzioni per il caricamento.

☆

Spedizione e termini. La videosigla dev'essere spedita alla IHT Gruppo Editoriale, IHT Video, Via Monte Napoleone 9, 20121 Milano entro il 20 giugno 1990. Devono essere inclusi i dati completi del mittente e il numero telefonico. L'autore della miglior sigla verrà retribuito con:

- Somma di 300 mila lire in contanti; abbonamento gratuito a 36 numeri di Commodore Gazette (del valore di 288 mila lire); collezione completa dei libri IHT attualmente in catalogo (del valore di oltre 600 mila lire).
- Presentazione della videosigla e del suo autore con un articolo sulla rivista Commodore Gazette.

☆

Lo stesso autore può inviare anche più di un prodotto. La videosigla vincitrice diverrà a tutti gli effetti proprietà della IHT che ne acquisirà il completo copyright. All'autore potrà essere eventualmente richiesto di apportare qualche modifica per venire incontro alle esigenze di produzione e questo s'intenderà compreso nel compenso sopra esposto. In tutte le produzioni della IHT Video in cui verrà utilizzata la videosigla apparirà il nome dell'autore nei titoli di testa o di coda. Il materiale anche se non prescelto, non verrà restituito.

LA POSTA DELLA GAZETTE

LA VOCE DEI NOSTRI LETTORI

UNA LAUREA PER L'AMIGA

Vi scrivo per porvi alcuni quesiti. Sono da poco possessore di un A2000, comprato per aiutarmi nella stesura della mia tesi di laurea. Dopo i primi entusiasmi sono arrivati molti dubbi e perplessità. Infatti a questo magnifico computer vanno affiancati programmi specifici al compito per cui il computer è stato comprato. Questa scelta è per me difficile, vista la moltitudine dei programmi esistenti e la scarsa professionalità dei venditori. Vi sarei perciò grato se vorreste segnalarmi un word processor, un foglio elettronico e un programma generatore di grafici, che siano nello stesso tempo professionali e maneggevoli.

In secondo luogo, avrei bisogno di una stampante che abbini eleganza e chiarezza dei suoi caratteri con la possibilità di stampare i grafici.

Massimo Tabasso
Savigliano (CN)

Abbiamo pubblicato questa lettera per ripetere ancora una volta che non possiamo rispondere a questo tipo di domande. Chiedere di citare il nome di un programma "professionale e maneggevole" è come chiedere qual è il programma ideale. I molti programmi esistenti nei settori che Lei cita sono, ciascuno a suo modo, professionali e maneggevoli. Hanno però caratteristiche diverse e prezzi diversi, e possiamo soltanto invitare Lei (e tutti i lettori che hanno lo stesso problema) a consultare l'indice degli arretrati di Commodore Gazette *e a leggere con attenzione le molte recensioni (o "prove hardware" per quanto riguarda le stampanti) che abbiamo pubblicato nei 24 numeri della rivista.*

PAURA DI VIRUS

Sono un giovane utente Amiga, alle prime armi con un computer di così notevoli capacità. La prima cosa che mi è stata detta quando mi sono trovato tra le mani questa potente macchina, è stato di tenere i dischetti sempre protetti. In questo modo, oltre evitare di cancellare o danneggiare il programma, non correvo il rischio che vi entrasse qualche virus. È da questo momento che mi sono chiesto cosa fossero i virus, ma non ho ancora trovato una risposta precisa.

Ho chiesto anche ad alcuni miei amici che possiedono un Amiga da più tempo di me, ma neppure loro mi hanno saputo dire che cosa fossero. Ho pertanto deciso di rivolgermi a voi sperando che possiate darmi una corretta ed esauriente risposta.

Marco Puliti
Sassuolo (MO)

La risposta è semplice e complessa al tempo stesso: un virus è un programma contenuto nei primi blocchi (quelli detti "di boot") dei dischi che eseguono lo startup. Il programma si carica automaticamente nella memoria del computer, dopodiché esegue varie operazioni che vanno dalla semplice copia di se stesso su tutti i dischi "non protetti" inseriti nel disk drive fino a distruttive operazioni che rendono inutilizzabili i dischi stessi (dipende dal virus). Per una spiegazione più accurata le consigliamo la lettura di due articoli: "L'AIDS del computer" (Commodore Gazette *n. 7/87) e "Sconfitta l'AIDS dell'Amiga"* (Commodore Gazette *n. 2/88). Quest'ultimo propone anche il listato di un efficace programma antivirus.*

Indirizzate tutta la corrispondenza per la rivista a:

COMMODORE GAZETTE
La posta della Gazette
Via Monte Napoleone, 9
20121 Milano

Preghiamo i lettori di essere concisi e concreti, per darci modo di rispondere al più grande numero possibile di lettere. La redazione si riserva comunque il diritto di sintetizzare le lettere troppo lunghe.

UN "TURBO" NELL'AMIGA

Chi vi scrive (e per la terza volta) è un vostro affezionato lettore, infatti

vi seguo dal lontano n. 1 del febbraio 1986, e se pur con tante tribolazioni per reperire i vostri numeri (vedi problema: in quale mese uscirà *Commodore Gazette*?), vi ho sempre seguiti con entusiasmo, a volte sfidando anche le ire del mio edicolante assillato dalla fatidica domanda: "È arrivata *Commodore Gazette*?". Ho sempre seguito con interesse le vostre recensioni sui programmi e sulle varie schede che si possono applicare all'Amiga (a proposito, posseggo un Amiga 500), anche se non sono alla mia portata almeno le schede per rendere compatibile un Amiga a un IBM, quindi ho sempre preso poco in considerazione il fatto che nel mio piccolo Amiga potessi avere un PC IBM compatibile. Ed ecco il vostro "bug", come si dice in gergo, infatti 20 giorni fa ho avuto da alcuni amici una copia di un programma che neppure sapevo che esistesse ed è il *Turbo Transformer 1.23*: immaginate il mio scetticismo quando, dopo averlo lanciato, il programma mi chiedeva il dischetto del DOS. Ho pensato tra me, questa è tutta scena; ma, fortunato me che possedevo un dischetto DOS per IBM e dopo averlo inserito ed aver incrociato le dita... meraviglia delle meraviglie... è apparso il PROMPT del DOS.

E, per farla breve, ho provato a lanciare e ad usare, cosa molto più importante, il *DATA BASE III Plus* ed il *Word Star 2000* e vi devo dire che oltre la lentezza, i programmi hanno funzionato benissimo. Ed eccoci allora allo scopo della mia lettera: è possibile che a voi sia sfuggito un programma del genere, perché se non ricordo male, voi non ne avete mai parlato, e se così fosse questa è una grossa ingiustizia nei confronti di chi non possiede un Amiga 2000 o di chi non ha soldi per comprarsi le schede di compatibilità. Certamente sono d'accordo con voi che il programma, così com'è, forse non funzionerà al cento per cento, ma signori miei, permettetemi di dire che è una grossa soddisfazione almeno da parte mia far "crepare" d'invidia chi ha un computer MS-DOS, ma non possiede la mia grafica.

A voi adesso la difesa.

Matteo Vizzani
Manfredonia (FG)

Anche i lettori più affezionati possono avere un attimo di... distrazione. Le segnalo l'articolo "La compatibilità IBM per Amiga: SideCar e Transformer*" pubblicato nel n. 4/86 di* Commodore Gazette, *in cui si parla del* Transformer *e dei suoi molti limiti. Se non siamo più tornati sull'argomento, è soltanto perché – come regola – parliamo soltanto di prodotti disponibili sul mercato, e il* Transformer *(che è un prodotto Commodore) è ormai fuori listino.*

L'ACCENTO AL POSTO GIUSTO

Vi scrivo per l'articolo "Amiga: una tastiera senza più segreti" (*Commodore Gazette* n. 1/90. Infatti, dopo aver fatto i salti di gioia perché il signor Palma Alessio era finalmente riuscito a domare il ribelle segno di elevamento a potenza, mi sono accorto che forse aveva peccato di eccessivo zelo.

Mi spiego: il fatto che non comparisse ^ dopo aver premuto SHIFT/ì era sicuramente un baco della mappa, ma non lo sono i simboli associati ai tasti F, G, H, J, K. A questi tasti, infatti, sono collegati cinque accenti diversi ´ ` ^ ~ ¨: ma a cosa servono degli accenti se non possono accentare le vocali? A nulla. Ma provate a premere ALT/K (simbolo associato: ¨) e premete subito dopo, ad esempio, la vocale "o". Magia!... vedrete comparire la "ö" con dieresi.

Ovviamente il "trucco" funziona solo con le vocali, con l'unica eccezione dell'accento circonflesso ^, che accenta anche la "n" (la "gn" spagnola). Ecco svelato il mistero dei caratteri fantasma: se sono seguiti da vocali, vengono fusi con queste ultime per creare vocali accentate, e il famigerato $08, penso, serve proprio a questo.

Valerio Peretti
Legnago (VR)

PROGRAMMATORE!

Sono un ragazzo di 16 anni e vorrei diventare programmatore. Vorrei sapere, avendo a disposizione un Amiga 500 espanso a 1 MB e tre libri, *Il manuale dell'AmigaDOS*, *L'Amiga* e *Programmare l'Amiga Vol. II*, quali altri libri devo comprare, oltre al prossimo *Programmare l'Amiga Vol. I*, visto che tutti questi libri non insegnano a programmare l'Amiga. Quali altri dispositivi devo comprare?

Spero che continuiate a parlare di musica.

Andrea Bellini

Se non ha mai programmato, dovrebbe iniziare acquistando un libro sul linguaggio C per principianti (non è necessario che sia specifico per l'Amiga). Le consigliamo quindi di rivolgersi a una libreria che abbia un vasto settore informatico e di chiedere un libro sul linguaggio C. Ce ne sono diversi e di diverse case editrici. Con quella base, e procurandosi un compilatore come il Lattice *o l'*Aztec, *potrà cominciare a utilizzare al meglio i nostri libri per programmare con l'Amiga.*

DOV'È FINITO LASERTERM?

Ho letto con molto interesse la vostra rivista che ho acquistato per la prima volta quest'anno e la ritengo molto buona, ricca di articoli e di novità e soprattutto con poca pubblicità. In particolare, ha destato molto interesse in me l'articolo sulle comunicazioni con un BBS; tra l'altro ho letto che con un particolare programma, il *Laserterm*, è possibile rendere l'adattatore telematico Commodore 6499 quasi come un vero modem.

Poiché sono in possesso del suddetto adattatore e m'interesserebbe acquistare il programma, potreste darmi qualche informazione più dettagliata sul programma ed eventualmente indicarmi a chi rivolgermi per acquistarlo?

Vincenzo Del Monaco
Napoli

Il programma Laserterm *è stato recensito sul numero 4/88 di* Commodore Gazette. *Purtroppo, l'indirizzo della casa produttrice non è più valido, visto che la LogOn Technology è scomparsa nel nulla, lasciandosi alle spalle anche alcune fatture inevase per pubblicità acquistata sulla nostra rivista. Le possiamo soltanto consigliare di rivolgersi a club o associazioni di appassionati, magari tramite un'inserzione su Classified.*

■

SOFTWARE AMIGA

A-Talk III 225.000
Absoft A/C Basic 490.000
Absoft A/C Fortran 745.000
Acquisition 1.3 620.000
Aegis Animator+Images 199.000
Aegis AudioMaster II 199.000
Appetizer 75.000
Aegis Impact 175.000
Aegis Sonix 159.000
Aegis Video Titler 275.000
AmKit 89.000
Ami-Alignment System 99.000
AniMagic 199.000
Animation Stand 99.000
Apprentice Lib Blocks 45.000
Apprentice Lib Flipper 45.000
Apprentice Lib Geometrics 45.000
Arexx 115.000
Ashas Caligrafonts 149.000
AssemPro 175.000
Award Maker Plus 99.000
Aztec C Professional 325.000
B.A.D. Disk Optimizer 99.000
BBC Emulator con cavo 129.000
BBS-PC 315.000
Becker Text 299.000
Benchmark C Library 225.000
Benchmark IFF Library 225.000
Benchmark Modula 2 425.000
Benchmark Simp Library 225.000
Bridge 5.0 75.000
Board Master 235.000
Butcher 89.000
C-Lite 125.000
ComicSetter Fun Figures 50.000
ComicSetter Sci-Fi 50.000
ComicSetter SuperHeroes 50.000
Choromap 120.000
Calligrapher 225.000
Climate 99.000
Comic Setter 125.000
Critics Choice 375.000
CygnusEd Prof. 225.000
DG Calc 99.000
DOS-2-DOS 125.000
Data Retrieve 175.000
Data Retrieve Prof. 625.000
Day By Day 75.000
Deluxe Music Co. Set 175.000
Deluxe Paint III 199.000
Deluxe Video III 185.000
Deluxe Photo Lab 175.000
Deluxe Production 349.000
Deluxe Print+Dpaint II 175.000
Descartes 75.000
Design 3D 199.000
Designasaurus 99.000
Desktop Budget 99.000
Devpac Amiga 149.000
Digipaint III 175.000
Digiworks 249.000
Disk Master 125.000
Disk-2-Disk 125.000
Diskwik 99.000
Dr. Drums 79.000
Dr. T Korg M1 335.000
Dr. T Copyist Prof. 599.000
Dr. T Keybd Cont Sys 499.000
Dr. T MIDI Rec. Studio 149.000
Aegis Draw 2000 550.000
Dynamic Drums 149.000
Dynamic Studio 375.000
E-Clips 239.000
Elan Performer 135.000
Extend 99.000
Encore 125.000
Excellence! 425.000
F.Sim Scenery #11 39.000
F.Sim Scenery #7 39.000
F.Sim Scenery #9 39.000
F.Sim Scenery Europe 39.000
FACC 2 75.000
Family Tree 99.000
Fancy 3D Fonts 125.000
FantaVision 99.000
FinePrint 99.000
Flight Simulator II 89.000
FlipSide 75.000
Flow 199.000
Font Set #1 75.000
Forms In Flight II 299.000
GFA Basic 165.000
Galileo 149.000
GoldSpell 2 99.000
Gomf 3.0 89.000
Gomf Button 185.000
Grabbit 75.000
Graphics Starter Kit 175.000
Hisoft Basic 199.000
Home Office Kit 370.000
Home Accounts 75.000
InterFont 249.000
Interchange Objects #1 50.000
Interchange 125.000
Jet 99.000
K-Gadget 75.000
K-Seka Assembler 125.000
K-Spread 149.000
Kara Fonts #1 99.000
Kara Fonts #2 149.000
Kara Sub Heads 149.000
Lattice C 5.0 Devl.Sys 750.000
Life Cycle 75.000
Lights, Camera, Action 165.000
Lion Fonts 149.000
Mailshot Plus 125.000
Masterpiece Fonts 425.000
MaxiPlan 500 299.000
MaxiPlan Plus 435.000
Medialine Backgrounds 89.000
Microfiche Filer 199.000
Microfiche Filer Plus 399.000
Modeler 3D 199.000
Movie Clips 50.000
Movie Setter 175.000
Music Module Starter 110.000
Music X 279.000
Newletter Fonts 75.000
Opus 1 MCL 249.000
P.A.S.E. 189.000
Page Flipper FX 349.000
Pagesetter II 249.000
Page Render 3D 349.000
Pic Magic 235.000
Photon Cell Animator 259.000
PixelScript 345.000
Transport Controller 499.000
Photon Paint 2.0 225.000
Ports Of Call 65.000
Power Windows 2.5 175.000
Prism Plus 149.000
Pro Video Plus 650.000
Pro Video Plus Font #1 250.000
Pro Video Plus Font #2 250.000
Pro Video Plus Font #3 250.000
Pro Video Plus Font #4 250.000
Pro-Board 499.000
Pro Data 219.000
Pro-Net 499.000
ProText 4.2 249.000
Professional Draw 349.000
Professional Page 1.3 650.000
Project D 99.000
Prowrite 2 225.000
Publisher's Choice 249.000
Quarterback 149.000
RubyView 249.000
Scribble Platinum 130.000
Sculpt 3D 215.000
Sculpt 3D XL 375.000
Sculpt 4D 1.350.000
Sculpt 4D Jr. 335.000
Sculpt Arch. Disk 69.000
Sculpt Future Disk 69.000
Sculpt Human Disk 69.000
Sculpt Interiors Disk 69.000
Sculpt Microbot Disk 69.000
Sculpt-Animate 3D 315.000
Sound Oasis 199.000
SoundTrax #1 39.000
SoundTrax #2 39.000
Structured ClipArt 115.000
Studio Fonts 75.000
Starter Kit 75.000
Studio Magic 225.000
Superback 135.000
Super DJ 55.000
Synthia 235.000
Templates 110.000
Terrain 50.000
Text Pro 140.000
The Director 150.000
The Director's Toolkit 89.000
The Works Platinum 420.000
Toolbox AmigaDos 125.000
Toolkit 99.000
Transcript 99.000
Turbo Silver 249.000
Turbo Silver Mod Pak 75.000
TV Graphics 110.000
Tv-Show 225.000
Tv-Text Professional 349.000
Typing Tutor 75.000
Ultra Card Plus 225.000
Video Effects 3D 399.000
Video Generic Master 175.000
Video Wipe Master 175.000
VideoScape 3D 399.000
VideoScape 3D Addendum 89.000
Videoscape Arch. Disk 69.000
Videoscape Future Disk 69.000
Videoscape Human Disk 69.000
Videoscape Microbot Disk 69.000
Virus Infection Protection 99.000
Who What When Where 89.000
WordPerfect 4.1 550.000
WordPerfect Library 275.000
Wshell 99.000
X-Cad Designer 299.000
X-Cad Professional 1.150.000
Xcopy II 75.000
Zoetrope 279.000

LIBRI AMIGA

Amiga C Adv Progr 49.000
Amiga Drive Inside&Out 72.000
Amiga Machine Language 38.000
Amiga System Prg Guide 85.000
Bantam AmigaDos Man II 52.000
Amiga ROM Kernel Manual 83.000
Amiga Hardware Manual 63.000

HARDWARE AMIGA

AMAS Sound Digitizer 299.000
Hard disk A-590 899.000
Espansione 2 MB per A-590 399.000
Mac-2-DOS con drive 950.000
SummaSketch Plus 18x12 1.690.000
SummaSketch Plus 1.150.000
Espansione 2 MB A-2000 799.000
ScanLock Vidtek 2.650.000
DigiDroid 175.000
DigiView 4.0 450.000
Drive esterno con switch 199.000
Drive esterno TrackDisplay 259.000
Drive esterno 5"1/4 275.000
Flicker Fixer 990.000
Future Sound 275.000
Hardcard 30 MB A-2000 999.000
Hardcard 60 MB A-2000 1.499.000
Drive interno A-2000 179.000
Midget Racer 68020 850.000
Midget Racer 68881/16 1.390.000
MiniGen 345.000
Perfect Sound 225.000
Scanner A4 1.495.000
A-Max con ROM 799.000
A-Max senza ROM 399.000
Fat Agnus 8372A nuovo ECS 149.000

SOFTSERVICE

Servizio importazione diretta di software originale, hardware professionale e libri di informatica.

SUPERPIC

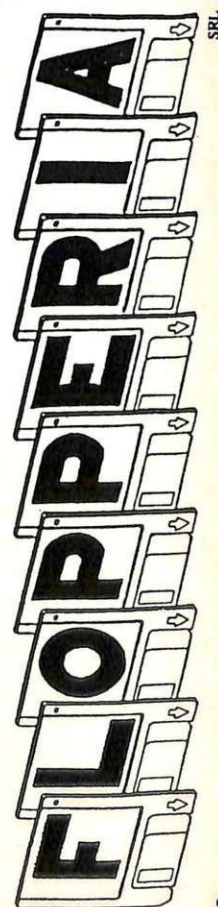
Digitalizzatore in tempo reale per Amiga 500 o 2000 con almeno 1 MB Ram, comprende un circuito Framestore ed anche un **genlock**. Digitalizza con 32.768 colori che sono convertiti nel modo HAM 4096 colori di Amiga con una sofisticato algoritmo di ottimizzazione. Inoltre funziona in overscan ed **elimina lo sfarfallio in alta risoluzione.**

1.590.000

SYNCRO EXPRESS

Eccezionale novità per Amiga: è finalmente disponibile il primo **copiatore hardware** per i dischetti Amiga! Con una speciale interfaccia collegata a 2 disk drives (quello interno al computer ed uno esterno), effettua copie di sicurezza, perfettamente funzionanti, di **qualsiasi software protetto** in meno di 50 secondi, compresi gli "impossibili" come Dragon's Lair.

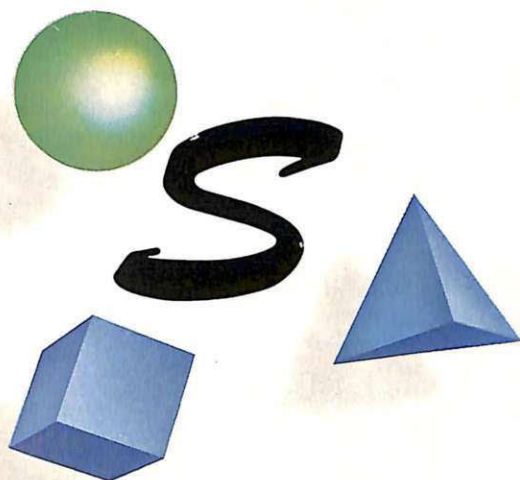
99.000

Prezzi IVA compresa

Viale Monte Nero 31
20135 Milano
Tel. (02) 55.18.04.84
(4 linee ric. aut.)
Fax (02) 55.18.81.05 (24 ore)

Negozio aperto al pubblico tutti i giorni dalle 10 alle 13 e dalle 15 alle 19.
Vendita per corrispondenza.
Sconti per quantità ai sigg. Rivenditori.

Non basta scrivere di essere seri professionisti per esserlo.

SOFTWARE GALLERY

UNA GUIDA PER ORIENTARSI NEL MONDO DEL SOFTWARE

INDIANA JONES AND THE LAST CRUSADE

Per la sua ultima avventura, l'archeologo più famoso del mondo si fa addirittura in due

Computer: C-64/128/Amiga (gioco d'azione) – Amiga (avventura grafica)
Supporto: Cassetta/Disco
Prezzo: L. 15.000/25.000 (gioco d'azione) – L. 59.000 (avventura grafica)
Produzione: Lucasfilm
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Il mondo del cinema è sempre stato un'ottima fonte d'ispirazione per quello dei videogiochi, al quale lo legano molteplici aspetti. Negli ultimi dieci anni sono stati realizzati innumerevoli programmi basati sulle pellicole più disparate, da *Robocop* (la versione videogioco è rimasta ai primi posti nelle classifiche di vendita per più di un anno) fino ai film muti di Charlie Chaplin e di Laurel & Hardy. Naturalmente tutti gli eroi della scuderia Spielberg/Lucas hanno avuto le loro versioni computerizzate, ma il record di presenze va all'avventuriero Indiana Jones, apparso in tre diversi giochi d'azione e in un paio di avventure, in una pittoresca varietà di formati.

Contro ogni logica, tuttavia, l'incredibile archeologo fanfarone non ha mai riscosso un grande successo quando è stato trasformato in eroe digitale: gli è andata molto meglio, per esempio, come protagonista di board-game e giochi di ruolo, o come pupazzetto di plastica.

Forse per combattere questa refrattarietà del mercato dei videogiochi nei confronti del personaggio, in occasione del film *Indiana Jones e l'ultima crociata* la Lucasfilm ha deciso d'invadere gli scaffali dei negozi con ben due versioni giocabili del film. Vediamole entrambe.

Il gioco d'azione. Disponibile sia per l'Amiga che per il C-64, questa versione ricade nella categoria dei giochi di piattaforme, ma senza riuscire a conquistarsi una posizione di preminenza, anzi.

Il gioco è diviso in quattro livelli che ripercorrono abbastanza fedelmente altrettante scene del film. Il primo quadro vede il giovane Henry Jones junior all'interno di un complesso di caverne nelle cui profondità si cela la perduta Croce di Coronado, risalente all'epoca dei conquistadores europei. Manovrando il joystick si guidano i movimenti di Indy facendolo camminare, saltare, ac-

SCHEDA CRITICA

INSUFFICIENTE
Un pessimo prodotto che non merita nessuna considerazione.

MEDIOCRE
Il programma ha alcuni difetti di fondo, anche se nel complesso raggiunge quasi la sufficienza.

DISCRETO
Un prodotto accettabile, ma non aspettatevi grandissime emozioni.

BUONO
Raccomandato vivamente: tra i migliori programmi della sua categoria.

OTTIMO
Eccezionale! Fino ad oggi non si era mai visto niente del genere.

cucciarsi, salire e scendere le provvidenziali funi che abbondano un po' ovunque.

Le caverne ospitano inoltre numerosi personaggi impegnati nella ricerca del manufatto, e tutti cercano di eliminare il ragazzo con pugni, colpi di accetta e persino di pistola. Per difendersi da questi biechi individui, Indiana può esibirsi in violentissimi uppercut o sfruttare le immancabili fruste che, raccolte durante il percorso, gli permettono per cinque volte di colpire gli antagonisti tenendosi a distanza di sicurezza.

L'esplorazione delle caverne riserva una lunga serie di spiacevoli sorprese fra le quali ricordiamo passerelle franante, pozzi profondissimi, stalattiti instabili, laghi sotterranei e creature misteriose e letali.. ma ognuno di questi ostacoli può essere aggirato o superato con un minimo di destrezza con i comandi. Quella che non può essere evitata è l'oscurità, che viene naturalmente tenuta a bada con le torce; ogni torcia ha purtroppo una durata limitata, entro la quale bisogna trovare una nuova fonte di luce. Se il nostro eroe viene sorpreso dal buio, infatti, perde una delle vite a sua disposizione.

Fra salti, spari, frustate e capocciate contro il soffitto, il livello termina con un balzo sul treno di un circo. Dopo una sequenza sui tetti del convoglio, che si può definire "d'intermezzo", il gioco prosegue con i labirintici corridoi delle catacombe di Venezia e con la scalata alla torre di un castello austriaco, sequenze virtualmente identiche alla prima, nelle quali cambiano solamente alcuni pericoli (compaiono topi, fulmini e trappole mortali al posto delle stalattiti e degli archeologi-rivali) e diventa possibile usare la frusta come una liana, per dondolarsi da una piattaforma all'altra.

Dopo un livello inutilmente complicato a bordo di un dirigibile Zeppelin (lo schema di gioco rimane ancora invariato) sul quale troviamo gli immancabili soldati nazisti, il gioco si conclude con un quadro leggermente diverso nel quale dobbiamo affrontare la famosa prova del "Verbo di Dio". Per chi non sapesse di che cosa stiamo parlando, spiegheremo che la sequenza è ambientata in una caverna il cui pavimento è costituito di una serie di lastre sulle quali sono incise alcune lettere dell'alfabeto. Muovendosi sulle lastre che, nell'ordine, formano la parola "IEHOVA" (ovvero "Dio", è possibile raggiungere la parte opposta della sala senza precipitare in un baratro senza fondo, recuperando così il Sacro Graal con il quale salveremo Henry Jones senior. L'unica difficoltà, in questo livello, è lo strettissimo limite di tempo che ci viene imposto per la soluzione, oltre che la stranissima prospettiva impiegata dal programma, che induce spesso a muoversi dove non si vorrebbe facendo così morire inutilmente il nostro personaggio.

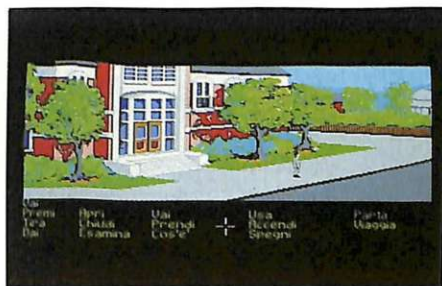
Il gioco, virtualmente identico nelle due versioni per C-64 e Amiga, non si può definire molto divertente: l'eccessiva difficoltà dei vari livelli rende ogni partita una frustrante agonia, mentre i

pochi giocatori che perseverano per vederne la fine sono destinati a rimanere delusi per la monotonia delle situazioni proposte e soprattutto per il pessimo finale, privo di una benché minima schermata grafica o sequenza speciale che riscatti il resto del gioco. Per questi problemi e la terribile e fastidiosa lentezza nella risposta dei comandi, non possiamo che consigliare ai nostri lettori appassionati di "platform game" di dirigere le loro mire altrove.

L'avventura grafica. Il secondo gioco ispirato a *Indiana Jones e l'ultima crociata* è senza dubbio molto migliore di quello che abbiamo or ora descritto.

Il programma segue la falsariga di due precedenti successi della Lucasfilm, *Maniac Mansion* e *Zak McKraken & the Alien Mindbenders*, certamente definibili come le migliori avventure grafiche mai pubblicate per home computer. Per chi non sapesse com'erano strutturati questi due giochi è necessaria una breve spiegazione.

Nelle avventure grafiche della Lucasfilm, lo schermo è diviso orizzontalmente in due fasce di uguale ampiezza: quella superiore ospita una schermata grafica e quella inferiore un testo. Ma c'è una grossa differenza rispetto alle comuni avventure semigrafiche: per l'inserimento dei comandi si usano esclusivamente il mouse o il joystick.

Ogni input da tastiera è completamente assente dal gioco, e il mouse muove un puntatore che svolge diverse funzioni a seconda dell'area di schermo in cui si trova. Nella parte grafica il puntatore serve infatti a dirigere i personaggi e a identificare gli oggetti, mentre in quella testuale permette di selezionare parole, opzioni di dialogo o oggetti presenti nel nostro inventario. Una normale sequenza d'azione comporta per esempio lo spostamento del protagonista nelle vicinanze di una bottiglia (perfettamente riconoscibile come tale dall'immagine che compare sul video), la selezione del verbo "PRENDI", il ritorno sulla schermata, la pressione del pulsante sulla bottiglia per indicarla come complemento oggetto e una nuova pressione per inviare il comando al computer. Sullo schermo si vede il nostro alter ego tendere una mano verso la bottiglia, soppesarla e mettersela in tasca. Subito dopo, l'oggetto "bottiglia" appare in un elenco situato sotto i verbi, dal quale lo potremo nuovamente selezionare con un semplice click del mouse.

Sopra: ha inizio l'avventura grafica per Amiga. Sotto: il gioco d'azione per C-64

Per quanto possa sembrare complicato, questo metodo di controllo si rivela immediatamente comprensibile e facile da impiegare. Ovviamente richiede una gran quantità di memoria per gestire le innumerevoli sequenze grafiche: per programmi di questo tipo, due o tre dischi sono la norma sia per il C-64 che per l'Amiga, e questo è anche il caso di *Indiana Jones and the Last Crusade*.

Ma veniamo al gioco vero e proprio. La trama del film viene riprodotta con una certa precisione (in alcune sequenze vengono riprodotte persino le carrellate e i dialoghi).

Sintetizzando una vicenda che molti di voi conosceranno bene, ricordiamo che la storia riguarda la ricerca del Sacro Graal, cui vengono attribuiti incredibili poteri taumaturgici. Indiana Jones è spinto in questa avventura dal desiderio di ritrovare e salvare il suo anziano padre, anche lui archeologo, che viene considerato la massima autorità mondiale a proposito del mitico calice.

La sorte e gli intrighi portano il professor Jones da Venezia all'Austria, e dalla Berlino del Führer alle rovine mediorientali di Iskenderun.

Per fortuna il gioco non si limita a seguire pedissequamente la sceneggiatura del film, ma vi aggiunge diverse sequenze e ne modifica altre, dando modo a chi ha già visto la versione cinematografica di rimanere ugualmente stupito dal dipanarsi degli eventi. Rispetto ai due precedenti titoli della Lucasfilm sono state aggiunte un paio di sequenze arcade controllate direttamente tramite il mouse, e la possibilità di fare a pugni con le immancabili SS sparse un po' per tutto il gioco.

Descrivere il programma senza citarne alcune scene è molto difficile, quindi ci limiteremo a segnalare alcuni aspetti delle scelte relative alla grafica e alla sceneggiatura. Nel primo caso citiamo la presenza di carrellate, stacchi, panoramiche, cambi di prospettiva e altri piacevoli cliché cinematografici dai quali non è escluso nemmeno il classico e melodrammatico «Nel frattempo, in un'altra parte del mondo...» seguito da un intermezzo animato. Per quanto riguarda la sceneggiatura, vanno segnalati gli innumerevoli tocchi umoristici e le citazioni di altri film o giochi contenute un po' dovunque: si va da una folle

serie di scritte sui muri in una biblioteca veneziana ad opera dei più grandi geni di tutti i tempi, alla presenza qui e là di personaggi e oggetti provenienti da *Zak McKraken & the Alien Mindbenders*, sempre osservati con una certa perplessità dal professor Jones.

Mentre grafica e sonoro non fanno testo (ignobile quanto involontario calembour), la vastità del gioco è tale da tenere anche i più assidui giocatori incollati al monitor per parecchio tempo, riservando loro una serie di piacevolissime sorprese. La cura posta nei particolari del programma fa di *Indiana Jones & the Last Crusade* il miglior titolo della già acclamatissima serie delle avventure grafiche firmate Lucasfilm. Sottolineamo ancora che la difficoltà ottimamente calibrata e mai esagerata rende il programma godibile anche per chi in genere non prende in considerazione questo genere di giochi.

Un ultimo elemento positivo è la traduzione in italiano del gioco, sia a livello di manualistica che di dialoghi e istruzioni visualizzati sullo schermo: gli strafalcioni e gli errori di grammatica purtroppo non sono rari, ma il programma guadagna comunque un ulteriore fascino per noi poveri latini, perennemente dimenticati dall'industria del software ricreativo.

Se non fosse per quest'ultima provvidenziale avventura grafica, l'archeologo di Lucas & Spielberg sarebbe probabilmente caduto in un oblio senza tempo, per quanto riguarda il mondo dei videogiochi. Non possiamo che lasciarvi consigliando senza riserve l'avventura grafica ai possessori dell'Amiga, mentre i "sessantaquattristi" faranno meglio a dimenticare le gesta di questo simpatico eroe, almeno sino al giorno di un improbabile nuovo gioco o di una conversione dell'avventura grafica.

F.R.

BLUE ANGELS

Né gioco, né simulazione: le acrobazie aeree continuano a mietere vittime

Computer: C-64/128
Supporto: Disco
Prezzo: L. 20.000
Produzione: Accolade
Distribuzione: C.T.O. (Via Piemonte 7/F, 40069 Zola Predosa – 051/753133)

Blue Angels fa parte di quella categoria di programmi che scatenano accese discussioni in redazione: c'è chi lo giudica un simulatore, chi un gioco, chi si rifiuta categoricamente d'inserire il disco nel drive. È una prova evidente che i nostri giudizi non sono e non possono essere "obiettivi" in assoluto, e che anche giochi bollati con un "insufficiente" potrebbero avere la loro parte di estimatori. Nel caso in questione, per avere un responso affidabile ci siamo rivolti a chi di aerei se ne intende per davvero, andando a chiedere l'opinione di un pilota e ingegnere aeronautico.

Dopo un paio di partite di prova, ci è stato detto che *Blue Angels* «riproduce abbastanza fedelmente le caratteristiche di un vero simulatore acrobatico, semplificandone tuttavia i controlli in modo da rendere il programma abbastanza giocabile». Impressionante.

Il problema è che, indipendentemente dal fascino che un programma di questo genere può avere per un professionista, noi semplici giocatori abbiamo trovato questo recente prodotto della Accolade un po' troppo eclettico: il tipico programma che rischia di deludere sia gli amanti delle "simulazioni" (a questo punto le virgolette sono d'obbligo) sia gli appassionati dei giochi d'azione. Ma vediamone in dettaglio la struttura.

Blue Angels si basa sulle gesta dell'omonima pattuglia acrobatica americana, che impiega gli F/A 18 per strabiliare gli appassionati delle evoluzioni aeree con imprese ai limiti dell'impossibile. Il programma permette di scegliere la posizione del proprio aereo all'interno della formazione e di modificare il livello di realismo del gioco alterando i parametri relativi allo scorrere del tempo e la quantità di strumenti di aiuto per il pilota.

Una volta impostato il realismo della simulazione, la scelta successiva riguarda il piano di volo: alcuni lunghissimi menu elencano le varie manovre che si possono affrontare singolarmente, mentre vi sono due opzioni specifiche per la simulazione di un'intera "esibizione" e per lo spettacolo vero e proprio, che si svolge volando a pochi centimetri dagli altri jet. Le selezioni sono accompagnate dalla comparsa di un diagramma tridimensionale di grande efficacia che mostra la rotta come una scia rossa all'interno di un corridoio trasparente a forma di parallelepipedo.

Dopo aver eseguito le varie selezioni, tutte tramite il joystick, ci si trova istantaneamente in volo, all'interno del proprio velivolo blu.

Lo schermo di pilotaggio è di tipo tradizionale, in soggettiva, e permette esclusivamente una visione frontale: le ormai diffuse visioni laterali e posteriori sarebbero sicuramente state superflue e addirittura controproducenti in un gioco come questo, dov'è importantissimo tenere sempre sotto controllo gli strumenti. E a proposito di strumenti, quello che colpisce di più in *Blue Angels* è proprio la loro estrema semplicità.

Al posto delle decine d'indicatori cui ci hanno abituato titoli come il leggendario *Flight Simulator II*, qui c'è un grosso display grafico centrale, affiancato da una coppia di cronometri e da un

paio di schermi che descrivono le manovre da effettuare. Gli altri strumenti, tra cui una bussola, diverse spie luminose e alcuni indicatori a barra, servono soltanto a fare scena: in realtà sono del tutto superflui ai fini del gioco.

Durante l'esecuzione delle manovre, comunque, i pochi strumenti che abbiamo elencato sono tutti necessari, senza alcuna eccezione. Lo svolgimento corretto di un'acrobazia prevede una precisa successione di letture; per darvi un'idea di come si svolge una partita con *Blue Angels*, ecco un esempio dei compiti che deve eseguire il "pilota".

La prima cosa da fare è leggere il display di destra, sul quale appare il nome della manovra da affrontare. Una parte vitale del gioco consiste nel memorizzare i movimenti di cui è composta la manovra, utilizzando le opzioni di allenamento sino a quando non si sono imparate quasi meccanicamente le combinazioni di joystick e pulsante necessarie. Spostando lo sguardo sul cronometro verde al di sopra del display centrale otteniamo un count-down che ci dice quanto tempo manca al successivo movimento della cloche, fondamentale per evitare di scontrarsi con gli altri aerei della formazione fittizia.

Nel momento in cui il conto alla rovescia raggiunge lo zero, l'indicatore grafico all'estrema sinistra del cruscotto indica, tramite a una serie di frecce luminose, i movimenti da compiere col joystick per eseguire la manovra. Memori di Pavlov seguiamo lo schemino ed ecco che la nostra attenzione viene richiamata sul display centrale, nel quale vediamo il corridoio di volo rappresentato da una serie di rettangoli in movimento. Al centro del corridoio spicca infatti il disegno di un aeroplano che il computer pilota lungo la traiettoria ideale; se il nostro mezzo se ne allontana, dobbiamo cercare di riportarlo a coincidere con quello del computer. Toccare le "pareti" del corridoio equivale a urtare il terreno o un altro jet, con risultati disastrosi.

Nella felice eventualità di essere sopravvissuti alla rocambolesca manovra, giunti a questo punto si torna sul count-down, che scorre inesorabile verso la successiva mossa della nostra elaborata acrobazia.

Questo è tutto. Visto da vicino, *Blue Angels* ricade nella triste categoria di quei "non-giochi" a cui appartengono anche *Dragon's Lair* e *Space Ace*, dove l'utente si limita a seguire le direttive del computer per riuscire ad arrivare in fondo alla partita. Fortunatamente, *Blue Angels* permette un certo margine di errore, anche se alla fine del volo il computer ci ammonisce a seguire più accuratamente le sue indicazioni, fornendoci una serie di grafici che ci fanno scoprire quanto siamo andati vicino alla morte durante lo show.

Il valore di questo programma è quantomai dubbio: da una parte c'è una struttura di gioco che rasenta la demenzialità, dall'altra il fatto che è stata resa molto bene la meccanica di un vero simulatore, anche a detta di un professionista. La mancanza delle sequenze di atterraggio e decollo influisce negativamente sulla qualità globale, ma ancora una volta va considerato che la Accolade ha voluto trasferire su computer soltanto la parte acrobatica di un volo, non quella delle manovre di routine che avrebbero complicato inutilmente il praticissimo sistema di controllo.

Gli amanti dei comuni videogiochi rimarranno senza dubbio delusi da questo programma: grafica e sonoro sono ridotti al minimo indispensabile, così come le azioni da compiere; i fanatici dei simulatori troveranno *Blue Angels* troppo simile a un "giochino" e torneranno senza troppi rimpianti ai loro *Falcon* e *F-29 Retaliator*.

In conclusione, ci sembra che il programma possa generare qualche interesse soltanto nella ristrettissima fascia degli appassionati di acrobazie aeree, che comunque non ci metteranno molto a scoprirne i segreti. Esisterà mai un bel simulatore di volo in formazione?

F.R.

IT CAME FROM THE DESERT

Ritornano i mostri degli anni Cinquanta

Computer: Amiga
Supporto: Disco
Prezzo: L. 49.000
Produzione: Cinemaware
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Negli anni Cinquanta le case cinematografiche statunitensi scoprirono la fantascienza. Una fantascienza hollywoodiana, ben lontana da *Metropolis* di Lang o dai drammi filosofici di *Zardoz*. I primi avvistamenti di oggetti non identificati nei cieli, la trasposizione cinematografica della *Guerra dei mondi* e l'atmosfera paranoide dell'epoca, in cui il sogno americano sembrava minacciato da ogni direzione, germinarono e fecero proliferare il filone delle "invasioni".

I produttori cinematografici fecero allora assalire la Terra (o meglio, gli Stati Uniti d'America) da infinite razze aliene, mostriciattoli più o meno umanoidi e dischi, sigari, piatti e vari poligoni volanti. La risposta del pubblico a questo genere di spettacolo fu inizialmente tanto positiva da causare una vera e propria saturazione del mercato che diede inizio alla fase discendente del fenomeno.

Per catturare nuovamente l'interesse del pubblico, gli sceneggiatori esplorarono nuovi modi di affrontare il genere, giungendo infine a creare il filone dei "mostri giganti". Grazie all'abilità tecnica di veri e propri maghi degli effetti speciali come l'indimenticabile Ray Harryhausen e ai rapidissimi progressi delle tecnologie legate alla cinematografia, fu possibile portare sullo schermo creature indicibilmente orrende, rese ancora più temibili dalle loro dimensioni colossali. I mostri in questione erano in realtà soltanto innocui e comunissimi animaletti (ragni, formiche, farfalle, seppie, roditori vari) o, nel caso di budget particolarmente consistenti, pupazzi animati "a passo uno" non più alti di una ventina di centimetri.

Il genere dei mostri giganti riscosse un successo incredibile, dando origine a una serie di pellicole che parevano uscire tutte dalla stessa catena di montaggio: i titoli originali suonano come *Il mostro venuto dal mare*, *Il mistero dell'abisso*, *Tarantola*, *La mantide omicida*, *L'incredibile Uomo-Colosso*, sottolineando sempre in maniera più o meno esplicita la natura della minaccia che attendeva lo spettatore. La trama rimaneva perlopiù costante, con un eroe maltrattato da tutti e considerato pazzo che tenta di avvertire l'incredula popolazione di un'anonima città dell'arrivo più o meno misterioso del mostro di turno, di cui lui solo è a conoscenza. Alla fine, non essendo creduto, parte lui stesso al salvataggio. Parallelamente a questi film si collocano i vari eredi di *Godzilla*, in cui il mostro viene combattuto in un ambito metropolitano, sfruttando tutte le forze (armate) a disposizione.

It Came From the Desert ricrea lo stile di quelle pellicole indimenticabili sfruttando spietatamente ogni clichè disponibile, seguendo e superando l'esempio di *Rocket Ranger*, altro famoso titolo della Cinemaware. La vicenda, introdotta da una meravigliosa sequenza animata e parlata, si svolge in un villaggio minerario battezzato "Fiato di Lucertola" (Lizard Breath), un ammasso di baracche in mezzo al deserto, nei pressi del quale precipita un meteorite radioattivo. Il giocatore impersona il geologo Greg Bradley, il "tipico" maschio californiano dal fisico perfetto e dal cervello finissimo.

Greg si reca al villaggio per studiare il meteorite, ma non appena arriva sul posto scopre l'esistenza di una razza mutante di formiche giganti intenzionate a sfruttare le loro nuove, colossali dimensioni in maniera piuttosto bellicosa. Da quel momento, il dottor Bradley dovrà fare l'impossibile per convincere la popolazione di Lizard Breath dell'esistenza dei formiconi antropofagi e quindi dirigere la Guardia Nazionale in una battaglia all'ultimo sangue contro i mostruosi insetti mutati.

Come gli altri prodotti della Cinemaware, anche *It Came From the Desert* si svolge tramite una serie di sottogiochi separati, riuniti fra loro da una schermata comune che in questo caso è costituita dalla planimetria del villaggio, nel quale ci muoviamo semplicemente puntando i vari edifici con il mouse. Poiché l'invasione delle formiche segue un ordine cronologico preciso e immutabile, lo scopo reale del gioco è di migliorare sempre di più i tempi in cui vengono affrontate le varie sequenze, riuscendo a preparare una controffensiva efficiente nel minor tempo possibile.

Vediamo i sottogiochi in una rapida panoramica.

Incendio: un po' a causa della temperatura esterna, un po' per merito di alcuni elementi locali che interferiscono nella lotta contro le formiche, capita che si debba affrontare una fase in cui bisogna fermare un incendio dilagante, con l'unico ausilio di un estintore a schiuma. La difficoltà sta tutta nel riuscire a identificare e controllare i focolai più pericolosi senza perdere tempo e schiuma preziosi in zone trascurabili dell'incendio.

Duello in auto: siamo nel periodo di *Gioventù bruciata* e di James Dean... volete che non ci siano dei ragazzotti fanatici che vagano per i lunghi rettilinei del deserto sulle loro auto superelaborate? Quando Greg incontra queste bande di teppisti, è costretto a partecipare a uno sport tutto particolare che si svolge dirigendo due auto una contro l'altra sino a quando uno dei due guidatori non si lascia prendere dal panico e "scarta", dirigendo l'auto fuori dalla strada. Qui non occorre un'abilità particolare, ma solo sangue freddo e una buona dose di fortuna.

Duello a coltellate: la vita nel deserto logora i nervi e genera turbe mentali, tanto che alcuni elementi sono soggetti a crisi di furia esplosiva per un nonnulla, e la cosa si risolve in genere a coltellate. Questa sequenza di grande valore educativo è molto simile alle scazzottate di *Rocket Ranger*: più che l'abilità con il joystick conta il colpo d'occhio e l'intuito necessario a cogliere impreparato l'avversario evitando nel frattempo i suoi colpi.

L'Ospedale: affrontando con qualche incertezza le tre scene descritte e alcune altre, si subiscono danni fisici che ci fanno ricoverare nell'ospedale cittadino, una struttura labirintica popolata da medici e infermieri da barzelletta. Per evitare di perdere del tempo prezioso, possiamo tentare una fuga di soppiatto, dribblando il personale e sfruttando a nostro vantaggio sedie a rotelle e ascensori. Si tratta di uno degli schermi più divertenti, grazie all'atmosfera frenetica dell'inseguimento, adeguatamente corredata da una discreta grafica e da effetti sonori.

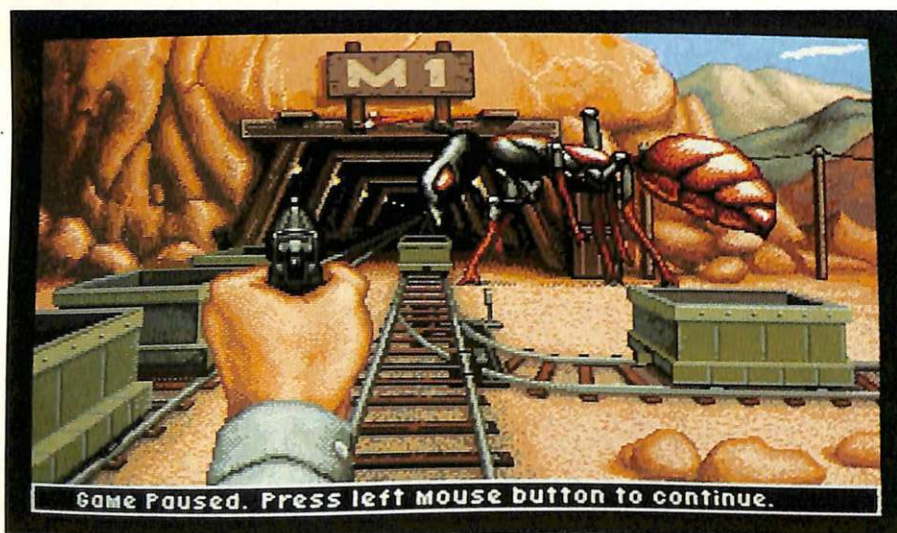
Perlustrazione e disinfestazione: Lizard Breath, come tutte le cittadine perse fra le dune, possiede un aeroporto dove possono decollare e atterrare piccoli mezzi da turismo. Greg può tentare di guidare uno di questi aeroplani in un giro di ricognizione sul perimetro cittadino, ottenendo così un resoconto aggiornato sullo stato delle forze mutanti. Equipaggiando adeguatamente il velivolo è anche possibile irrorare il terreno di pesticidi, sopprimendo gran parte degli avversari. Naturalmente non è così facile come sembra. L'aereo non è facilissimo da controllare e ha una scarsa autonomia: basta una manovra sbagliata, e finire all'ospedale è inevitabile.

Incontro diretto: durante le sue

peregrinazioni fra le sabbie del deserto, Greg rischia ovviamente d'incappare proprio in una delle formiche sovradimensionate, frutto della contaminazione radioattiva. In questo caso lo schermo cambia per mostrarci una visione in soggettiva del circondario, dominato dalla creatura a sei zampe: guidando una realistica mano armata di pistola bisogna riuscire a centrare le antenne del mostro (suo unico punto debole) prima che decida di assalirci.

Oltre alle spettacolari sequenze ni, il programma ci impone un'ultima, pericolosissima missione: penetrare nel superformicaio da cui provengono le creature mutanti, trovare la loro regina e piazzare esplosivi in quantità sufficiente da porre fine alla minaccia in modo definitivo... cercando di non rimanere intrappolati nei tunnel.

It Came From the Desert è sicuramente uno dei migliori titoli della produzione Cinemaware. I fan di questa software house vi troveranno parecchie ore di diverti-

elencate, *It Came From the Desert* offre un gran numero di opzioni prettamente strategiche che permettono d'interagire con i pigri e sinistri abitanti del villaggio minerario, di compiere studi scientifici sul fenomeno della mutazione, di comunicare con amici e colleghi rimasti a casa e, nelle sequenze finali, di guidare la rappresaglia militare contro i mostri. In questa fase, *It Came From the Desert* si trasforma in un misto di war-game e di gioco d'azione, in cui le scelte tattico-strategiche sono importanti quanto la capacità di prendere la mira con i propri pezzi d'artiglieria.

Supponendo di essere riusciti a respingere l'attacco dei formico-

mento, mentre chi sinora era rimasto scettico di fronte a giochi come *Rocket Ranger* o *Defender of the Crown*, graficamente stupendi ma forse un po' limitati sul piano della giocabilità, avrà modo di ricredersi. Dedicato a tutti i nostalgici degli anni Cinquanta e ai più giovani, che non hanno avuto modo di vivere il tempo in cui i mostri marciavano sulle città.

Un ultimo particolare: a differenza della maggior parte dei giochi prodotti sino ad oggi, il nostro personaggio non muore mai. Di qualunque episodio spiacevole si trovi a essere protagonista al massimo finisce all'ospedale. Una tendenza positiva...

F.R.

CHAMBERS OF SHAOLIN

Le arti marziali tornano su computer grazie alla Thalion

Computer: C-64/128/Amiga
Versione: Amiga
Supporto: Cassetta/Disco
Prezzo: L. 15.000/18.000
Produzione: Grandslam (Thalion)
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Negli ultimi anni, in molti Paesi d'Europa abbiamo assistito alla nascita di agguerrite software house specializzate in giochi, che stanno cominciando a intaccare il monopolio anglosassone (USA e Inghilterra) in questo settore.

La Thalion è la più giovane delle case di software tedesche, e nella presentazione alla stampa che ha avuto luogo un paio di mesi fa è già riuscita a catturare l'interesse di molti operatori del settore annunciando alcuni titoli veramente impressionanti sia per il C-64 che per l'Amiga. Il primo prodotto marchiato Thalion era quindi atteso con impazienza, e non appena è arrivato sulla nostra scrivania è stato caricato nel disk drive senza indugio.

Chambers of Shaolin è ambientato nel famoso monastero cinese in cui si narra siano nate le arti marziali, in un anno imprecisato del passato. Il protagonista è un aspirante karateka, tale Hang Foy Qua, giunto al tempio per imparare le tecniche di combattimento che gli permetteranno di liberare il proprio villaggio dallo strapotere delle milizie imperiali.

Il programma è diviso in due sezioni principali, la seconda delle quali non può essere giocata senza che prima sia stata completata la precedente. Nella prima sezione, Hang Foy Qua viene infatti sottoposto alle sei prove delle Camere di Shaolin, grazie alle

quali acquisirà le tecniche che gli serviranno in battaglia; nella seconda fase l'addestramento è terminato, e il karateka affronta una serie di nemici per liberare il villaggio. Vediamo adesso i vari quadri nell'ordine in cui si presentano.

La prima camera ospita la Prova del Bastone, ovvero un sottogioco a tempo nel quale il nostro rappresentante su schermo deve evitare i colpi di un agguerrito monaco armato di *bo* (bastone lungo). La difficoltà sta tutta nel capire a quale altezza verrà sferrata la bastonata successiva, e scegliere di conseguenza la mossa giusta fra le tre a nostra disposizione. La sezione termina allo scadere del tempo o, più spesso, quando abbiamo ricevuto troppe bastonate.

La Prova dell'Agilità è quella che Hang Foy Qua deve affrontare nella seconda camera, un vasto salone arredato ovviamente in stile orientale. Qui veniamo bersagliati da una gran quantità di oggetti più o meno pericolosi (accette, vasi, palle fiammeggianti...) che dobbiamo evitare alla bell'e meglio, producendoci eventualmente in spettacolari giravolte sulle mani e altre simili amenità.

La terza camera è in realtà uno spazio aperto, anzi un laghetto, per essere precisi. Nel bel mezzo del lago si trovano quattro pali che vengono mossi meccanicamente dentro e fuori dall'acqua. Abbiamo a disposizione tre tentativi per riuscire a completare la Prova dell'Equilibrio, che ci vede saltare da un palo all'altro secondo un ordine ben preciso. Dato che non ci chiamiamo Remo Williams, l'impresa si rivela molto più ardua del previsto, grazie anche all'imprevedibilità del computer che a volte ci fa cascare a mollo senza motivo.

La Prova della Velocità è ambientata in un pozzo che si riempie d'acqua lentamente ma inesorabilmente. Per riuscire a fermare il liquido, Hang Foy Qua deve prendere a calci un grosso pendolo metallico (ahia!), che se arriva abbastanza in alto nelle sue oscillazione va a colpire un buffo meccanismo composto di corde, accette, carrucole e funi presente su entrambi i lati del pozzo. Riuscire nell'impresa richiede una coordinazione notevole, non tanto per colpire la sfera, ma per abbassarsi ed evitarla quando ricade verso di noi a tutta velocità.

Fortunatamente non è necessario portare a termine con successo la prova precedente per accedere alla quinta camera, altrimenti ben pochi riuscirebbero a vedere la Prova della Forza. Qui la situazione sconfina nel cliché: il nostro eroe giallastro deve infatti spaccare una serie di tavolette colpendole col taglio della mano. Per farcela, bisogna prima di tutto prendere bene la mira, e poi "caricare" il braccio con un frenetico movimento del joystick. Difficile quasi quanto nella realtà.

L'ultima fase dell'addestramento si svolge nuovamente all'aperto: è la Prova del Fuoco. Hang Foy Qua si trova su un ponte, mentre il suo mentore lo bersaglia con una serie di globi fiammeggianti. Oltre a cercare di non essere colpiti, dobbiamo evitare che sul nostro ponte si ammassi più di un certo quantitativo di globi, pena la fine dell'addestramento. Stranamente, l'ultima sembra essere la prova più facile della prima sezione.

Terminato l'addestramento, gli utenti Amiga hanno la possibilità di salvare la situazione su disco (non abbiamo potuto vedere la versione per C-64, ma il manuale esclude questa possibilità). L'idea è senz'altro lodevole, perché a questo punto siamo in possesso di un karateka dalle capacità che dipendono da come abbiamo affrontato le prove... e per superare la seconda fase è consigliabile avere a disposizione quanto di più simile a un Bruce Lee riusciamo a ottenere. Non ci metterete molto ad accorgervi che un allenamento andato ottimamente potrebbe non ripetersi una seconda volta.

La seconda parte del gioco segue la falsariga di molti altri programmi dello stesso genere: a ogni combinazione di joystick e pulsante corrisponde una mossa di karate più o meno tradizionale, che dobbiamo impiegare contro un avversario capace dei nostri stessi movimenti.

Le battaglie si svolgono contro uno sfondo di pagode, fiori di pesco e un'incredibile quantità di luoghi comuni sull'oriente, e rappresentano forse la parte peggiore del gioco a causa della loro scarsa spettacolarità. L'unico motivo d'interesse, in questa fase, è lo scontro col quinto avversario, una sequenza molto difficile da portare a termine, per la quale il manuale non è praticamente di nessun aiuto.

Dopo le roboanti presentazioni, *Chambers of Shaolin* non fa certo un grande onore alla Thalion. Anche se il programma non mostra difetti evidenti, la realizzazione rimane sempre un po' sotto le righe, dando al tutto un sapore di *deja vu* che non invoglia certo il giocatore. La simpatica idea di dare attributi variabili al nostro karateka si perde nel nulla di fronte alla scialba realizzazione dei combattimenti della seconda fase. Al momento di scrivere questa recensione, ripensando al programma, l'unica cosa che ci è subito tornata in mente è stata la serie di schermate di caricamento.

Chambers of Shaolin è senz'altro consigliabile a chi ancora non possiede un gioco di arti marziali, ma i possessori di *International Karate Plus* o prodotti similari non sentiranno certo il bisogno di aggiungere questo titolo alla loro collezione.

F.R.

SIM CITY

***Life*^{2} = $\sqrt{\textit{Populous}}$**

Computer: C-64/128/Amiga
Supporto: Disco
Prezzo: L. 28.000/59.000
Produzione: Infogrames
Distribuzione: C.T.O. (Via Piemonte 7/F, 40069 Zola Predosa - 051/753133)

La storia si ripete: lo abbiamo imparato dai libri di scuola, lo abbiamo spesso sperimentato di persona e adesso ritroviamo questo fenomeno anche nel mondo dei videogiochi, dove bastano pochi anni per veder tornare in auge mode e tendenze che si pensavano completamente esaurite.

Tanti, tanti anni fa, per esempio, i possessori di Apple II passavano le grigie giornate invernali davanti alle "colonie cellulari" di *Life*, rappresentate da miriadi di puntini che nascevano, morivano e si riproducevano secondo alcune semplici leggi sotto l'occhio vigile del calcolatore. Spesso, fra una coltura e l'altra veniva poi caricato il leggendario *Santi & Paravia*, primordiale simulazione commerciale ambientata fra borghi medioevali.

Oggi che gli Apple II sono molto meno diffusi, trovare quei programmi è diventata un'impresa degna di Indiana Jones, tanto che fior di riviste d'informatica continuano da decine di numeri la loro ricerca di un *Life* aggiornato, riveduto e corretto per i relativamente nuovi mostri a 16 bit. Mentre i programmatori nostrani continuano la loro appassionante ricerca dell'algoritmo perfetto, le software house d'oltreoceano sembrano aver pensato di rilanciare la moda delle simulazioni ambientali.

Il primo prodotto che ci viene in mente parlando di questo genere di programmi è naturalmente *Populous*, lo strano misto di gioco e simulazione creato dalla Electronic Arts, che ha ottenuto un'unanime e caldissima accoglienza da tutte le riviste del settore ed è stato persino preso in considerazione da alcune pubblicazioni non specializzate che in genere bollano il software, soprattutto quello ludico, come un prodotto incomprensibile, stupido e inutile, riservato a una ristrettissima élite d'individui in camice bianco (o in camicia di forza, a seconda dei casi).

Per quelle poche persone che non sanno di che cosa stiamo parlando, puntualizziamo che *Populous* è un programma nel quale i giocatori plasmano un intero mondo, generando picchi e scogliere, provocando terremoti e inondazioni, edificando capanne e palazzi e manipolando gli abitanti del mondo così creato.

Però la scala gigantesca degli eventi di *Populous* influiva negativamente sulla simulazione ambientale, e molti limiti li aveva anche il minuscolo microcosmo di *Life*, che diventava rapidamente noioso per la sua mancanza di varietà. Era necessario trovare una via di mezzo, e i programmatori della Maxis sembrano avere finalmente trovato il giusto compromesso con *Sim City*.

Il gioco riguarda le vicende di una città, una struttura sociale relativamente semplice eppure capace di evolversi in un agglomerato di notevoli dimensioni.

Giocando con *Sim City*, l'utente può creare dal nulla una metropoli che dovrà poi amministrare, considerando gli innumerevoli problemi economico-sociali inevitabilmente connessi a una simile struttura. Per un caso strano, le due versioni del programma (Amiga e C-64) presentano notevoli differenze, che ci obbligano a esaminarle separatamente.

Versione Amiga. Fino a qualche mese fa, la versione Amiga di *Sim City* si trovava in circolazione (stiamo parlando di copie d'importazione) in due formati diversi per i computer dotati di 1 MB o di soli 512K. Le differenze fra le due versioni erano limitate alla rifiniture grafiche (un particolare del tutto marginale in programmi di questo tipo), ma è comunque preoccupante pensare che i programmatori della Maxis abbiano avuto bisogno di più di 524.288 byte per il loro codice.

Caricato il programma, l'utente si ritrova di fronte a un terreno disabitato, comprendente foreste, fiumi, mari e pianure. La prima scelta da fare sta nell'accettare o meno la zona proposta: in caso negativo si può procedere alla creazione del proprio scenario impiegando un comune editor, o passare direttamente al caricamento di una situazione precedentemente salvata su disco. Fra gli scenari presenti nella confezione originale troviamo esempi di città di ogni genere: il piccolo villaggio rurale, la metropoli bisognosa di ristrutturazione, il traffico e lo smog di Tokio, l'artificiosità urbana di Brasilia.

In ogni caso, il giocatore passa a selezionare un livello di difficoltà che andrà a influire principalmente sul budget a disposizione del municipio e finalmente la simulazione ha inizio. Utilizzando un sistema di icone, l'utente può edificare strutture di diverso tipo sul suo terreno; le unità principali impiegate nella città comprendono quartieri industriali, zone residenziali e zone commerciali. Ogni tipo di zona è influenzata dalle altre due. Come è facile intuire, fabbriche, scuole e negozi non potranno funzionare senza personale e fruitori, che del resto non andranno mai a vivere in un luogo che non dia possibilità di trovare lavoro. Le interrelazioni socio-economiche di queste tre unità costruttive già farebbero la gioia di molti studenti di economia e sociologia, ma *Sim City* offre ben altro...

Nel menu troviamo per esempio le centrali elettriche (termoe-

lettriche o nucleari), fondamentali per l'esistenza di un centro urbano ma che portano con sé problemi d'inquinamento e fattori di rischio. Per favorire l'economia della città ci sono anche porti e aeroporti, mentre parchi, canali e ponti serviranno a migliorare la vivibilità del luogo.

L'aumento della popolazione ha come inevitabile risultato l'aumento della criminalità, e per questo il menu di *Sim City* permette di edificare una centrale di polizia; non manca la caserma dei pompieri, e nemmeno lo stadio.

Ogni unità costruttiva ha un certo costo, così come le strade, le ferrovie e le linee elettriche: nelle prime fasi della simulazione il problema sta proprio nel far quadrare il bilancio del municipio, che deve offrire il meglio ai propri cittadini in modo da aumentare l'immigrazione e di conseguenza l'incasso annuale dovuto alle imposte. Proprio le imposte sono infatti l'unica fonte di guadagno per il giocatore, che spesso si troverà a doverle aumentare per far fronte a emergenze particolari.

Emergenze che, nella fantasia un po' perversa dei programmatori, assumono un gran numero di forme. Possiamo suddividere in due categorie i problemi in cui è possibile incappare: le "piaghe sociali" e le "catastrofi più o meno naturali". Nella prima categoria troviamo eventi strettamente legati alla natura stessa della città: criminalità, inquinamento, disoccupazione e costo della vita. Ciascuno di questi fattori viene tenuto costantemente sotto controllo dal programma, che mette a disposizione in ogni istante una serie di grafici d'immediata comprensione, a cui si può accedere dal menu principale. Grazie a questi schemi e alle relative "mappe tematiche" è possibile ovviare con notevole precisione a ogni disagio, anche se spesso la cosa comporta una riprogettazione radicale della metropoli.

Un'interessante trovata è poi il "menu delle catastrofi", che aggiunge ulteriore difficoltà al tutto. Con un semplice sistema di icone e puntatori, infatti, è possibile scatenare incendi, terremoti e uragani sui poveri cittadini, che potranno così sperimentare l'efficacia dei piani d'emergenza previsti dalla città. Il programma consiglia infatti una serie di misure preventive d'interesse pubblico e sociale, quali la costruzione di un bacino idrico artificiale che possa essere impiegato come un gigantesco estintore in caso d'incendio e altre ardite operazioni urbanistiche. L'opzione più divertente e irreale, comunque, prevede la presenza di un gigantesco mostro "alla Godzilla" in città, che potrebbe essere divertente vedere girare per le strade di Tokio (una delle città fornite sul disco).

Sopra: la versione per Commodore 64.
Sotto: la versione per Commodore Amiga

Versione C-64. Ed ecco le note dolenti: benché *Sim City* in versione a 8 bit mantenga la struttura generale che abbiamo descritto, per qualche insondabile ragione una buona metà delle opzioni scompare.

Senza pompieri, polizia, statistiche sulla criminalità, centrali nucleari, ferrovie, stadio e opzioni relative, *Sim City* rimane certamente un programma simpatico e interessante, ma totalmente privo della varietà che lo renderebbe un gioco di buon livello. Fondare colossali megalopoli potrà anche essere divertente, ma già nelle prime partite ci è capitato di trovare un unico problema strategico: l'impossibilità di estendere l'area abitata oltre la mappa principale. Dopo qualche ora di gioco, infatti, ogni zona era stata trasformata in un incredibile agglomerato cyberpunk che si estendeva a perdita d'occhio in ogni direzione, senza però suscitare nella popolazione il benché minimo disagio. I casi sono due: o possediamo una sensibilità urbanistica eccezionale o il gioco è stato semplificato eccessivamente.

E parlando di struttura del programma, permettete un'ultima considerazione critica: benché *Sim City* sia senza dubbio completo, interessante e forse persino educativo (caratteristica completamente assente in gran parte dei videogiochi), la sua semplicità strutturale è evidente. Eccettuata la parte grafica, peraltro piuttosto dimessa, chiunque possieda una minima infarinatura di programmazione è senza dubbio in grado di scriversi il proprio *Sim City* in meno di una settimana, con poche centinaia di righe Basic.

Orbene, con queste premesse come si fa a giustificare l'esistenza di ben due versioni Amiga e di una per C-64 tanto mutilata? Sono stati del tutto dimenticati i tempi eroici in cui ci si destreggiava con i 3,5K del VIC-20? Per chi invece non vede motivo d'indignarsi di fronte a una situazione simile, *Sim City* (soprattutto sull'Amiga) può costituire un'interessante evasione dai temi trattati normalmente nei videogiochi.

F.R.

THE UNTOUCHABLES

Un piacevole spin-off dalla Ocean

Computer: C-64/128/Amiga
Supporto: Cassetta/Disco
Prezzo: L. 18.000/25.000/29.000
Produzione: Ocean
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Gli *intoccabili* è da citare tra i film di successo degli anni Ottanta: l'idea di riportare sul grande schermo una storia "di gangster" è piaciuta al pubblico e alla critica, anche se dell'opera di Brian De Palma è stata apprezzata soprattutto l'impeccabile realizzazione tecnica.

Ma siamo qui a parlare di videogiochi, quindi passiamo subito dal film allo spin-off. Nel velocissimo mondo del software, vedere pubblicato lo spin-off del film a circa tre anni di distanza è stata una vera sorpresa; forse è addirittura l'inizio di una nuova era. Dopo decine di programmi tratti da questo o quell'altro film a tempi di record e di conseguenza poco curati (vedi *Ghostbusters II*), in questo titolo si nota una piacevole cura dei particolari, che lo rende complessivamente più giocabile e divertente. C'è soltanto da sperare che l'esempio venga seguito.

Gli "intoccabili" erano, durante gli anni del proibizionismo, un gruppo di agenti speciali dediti alla lotta contro la criminalità organizzata. I tutori dell'ordine operavano in borghese e godevano di una totale immunità dalle leggi vigenti, che consentiva loro metodi operativi poco ortodossi ma d'indubbia efficacia.

Il gioco segue la trama del film, in cui un gruppo di agenti guidati da Elliott Ness tenta di abbattere l'impero di Al Capone, potentissimo boss americano divenuto oltraggiosamente ricco grazie a una vasta rete di attività illecite, prima fra tutte lo spaccio di liquori "proibiti". Il programma è diviso in sei fasi, ognuna delle quali viene aperta da una schermata introduttiva che aiuta a "entrare nell'atmosfera".

Versione C-64. Il primo quadro del gioco è ambientato in un magazzino di Chicago, nel quale gli uomini di Capone hanno un deposito di bottiglie. In questo livello si devono raccogliere le informazioni necessarie per sventare l'invio di una grossa partita di alcolici.

Se vogliamo evitare di leggere sul quotidiano locale la notizia della "morte in un bagno di sangue" del nostro eroe, dobbiamo eliminare i numerosi gangster che ci assalgono a colpi di mitraglietta o a mani nude, balzando fuori dalle pile di casse di cui il magazzino è pieno. Per difenderci abbiamo con noi nientemeno che un fucile a canne mozze, forse un po' lento da ricaricare ma di sicura efficacia.

Mentre cerchiamo di recuperare le dieci informazioni necessarie, non dobbiamo trascurare di raccogliere gli oggetti che i nemici uccisi lasciano cadere a terra. Possono essere fucili mitragliatori, rose che reintegrano l'energia

o bonus che aumentano il tempo a nostra disposizione. La struttura del gioco è tale che risolvere il quadro senza questi oggetti è un'impresa impossibile, per cui è importantissimo saper dividere il proprio tempo fra le attività di raccolta e quelle di "caccia".

Quando, dopo molte partite e dopo esserci arrampicati infinite volte sulle casse del magazzino, riusciremo a raccogliere le dieci informazioni, verremo finalmente a sapere che la spedizione si svolgerà attraverso il confine USA-Canada e che i camion dovranno passare lungo uno stretto ponte. Gli intoccabili preparano quindi un'imboscata sul ponte, nella speranza di distruggere il carico di liquori, ma come sempre si trovano di fronte gli agguerriti scagnozzi di Capone.

A differenza del primo quadro (che si giocava con una visione laterale), qui lo schermo è impostato su una prospettiva realistica che dà un notevole aspetto di tridimensionalità, e il nostro personaggio viene visto da dietro e dall'alto mentre, sdraiato a terra, rotola a destra e a sinistra per evitare i proiettili dei criminali e colpire le bottiglie di liquore che appaiono a caso sullo schermo. Il quadro termina quando riusciamo a distruggere tutte le bottiglie o quando finisce il tempo.

La terza sezione è probabilmente la più spettacolare: si basa su una visuale tridimensionale in prima persona della città di Chicago e ci vede correre verso la stazione ferroviaria nel tentativo di fermare uno dei contabili della malavita, in procinto di lasciare il Paese per sfuggire alla rete degli intoccabili. Mentre procediamo fra strade, piazze e vicoli, veniamo rallentati dai cecchini, che vanno eliminati con le solite armi da fuoco; il problema è che le armi vanno ricaricate, e per far questo dobbiamo perdere del tempo prezioso gettandoci in qualche vicolo laterale dov'è possibile anche cambiare personaggio. Grazie alle ottime animazioni, sembra quasi di sentire nell'aria l'odore della polvere da sparo, ed è un peccato che l'alta difficoltà dei livelli precedenti renda complicato raggiungere questa sezione.

Quando arriviamo alla stazione gli eventi precipitano, e ci tocca affrontare l'ennesimo scontro a fuoco, ostacolati per di più dal fatto di dover guidare la discesa di una carrozzina abbandonata sui gradini dello scalone. E, come sempre, dobbiamo badare a non colpire gli incolpevoli passanti. Questa sezione è forse la meno riuscita del gioco, sia per una certa macchinosità nei comandi sia per la grafica impiegata, primordiale a paragone di quella del quadro precedente.

Il quinto livello gioca tutto sulla velocità e sulla capacità di mira acquisita col joystick: l'ultimo gangster rimasto in vita ha infatti catturato il contabile, e minaccia di ucciderlo entro cinque secondi. Dobbiamo centrarlo al primo colpo in un punto vitale, altrimenti perderemo le informazioni utili per incriminare Al Capone.

L'ultimo quadro è ambientato sui tetti di Chicago, quando ormai Capone è stato arrestato ed è in attesa di giudizio. Ora dobbiamo bloccare il suo braccio destro Frank Nitty, che ha rilevato l'impero di Capone. Alle prese con il cronometro per l'ultima volta, dobbiamo inseguire e catturare Nitty colpendolo a morte con l'ultima arma rimastaci. Dopo quel che abbiamo passato, l'impresa non è poi tanto difficile, e se riusciamo a raggiungere questo quadro è pressoché certa la comparsa della tanto agognata pagina di giornale sulla quale viene annunciata la sconfitta della criminalità a Chicago.

Versione Amiga. Sul computer a 16 bit della Commodore le differenze sono minime, ma giocano tuttavia a favore del tanto criticato C-64. Per quanto grandi siano le capacità dell'Amiga, il gioco risulta più scialbo, difficile da controllare e calibrato decisamente male, con alcune sezioni praticamente impossibili da superare e altre troppo semplici. A livello tattico non sono riscontrabili grandi cambiamenti, mentre la mancanza dei ragtime presenti sul C-64 fa perdere al programma molta atmosfera, una cosa che ci sembra assolutamente ingiustificabile.

Il magazzino di Chicago del primo quadro nelle versioni Amiga (sinistra) e C-64 (destra)

The Untouchables, pur rimanendo fondamentalmente uno shoot'em up, è un programma molto gradevole che fa della varietà il suo punto di forza. La versione per C-64 possiede rispetto a quella per Amiga molta più atmosfera e tocchi di classe (un esempio su tutti il poster pubblicitario di "Tempi moderni" con Charlie Chaplin nel primo livello), ma su entrambi i computer c'è modo di passare parecchi giorni nel tentativo di sconfiggere Capone & C. Dategli un'occhiata.

F.R.

X-OUT

Il nuovo shoot'em up "clonato" della Rainbow Arts

Computer: C-64/128/Amiga
Versione: Amiga
Supporto: Cassetta/Disco
Prezzo: L. 18.000/29.000
Produzione: Rainbow Arts
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Qualche tempo fa, recensendo *Katakis* della Rainbow Arts, giungemmo alla conclusione che i programmatori di questa software house tedesca avessero indubbiamente delle capacità tecniche notevoli, ma mancassero completamente di originalità. Sia *Katakis* sia gli altri prodotti Rainbow Arts erano semplici copie di coin-op da sala giochi, con sprite leggermente modificati ma altrimenti puri "cloni" di *R-Type*, *Super Mario Bros*, *Penguin-kun Wars* e altri titoli più o meno conosciuti del variegato universo videoludico.

X-Out (pronunciato "Cross Out") non fa eccezione alla regola. Alla Rainbow Arts hanno pensato bene di prendere due dei giochi più graditi del momento, fagocitarne gli elementi migliori e mischiarli in un unico prodotto tecnicamente sbalorditivo ma comunque vicino al plagio. Diremo subito che gli originali vampirizzati sono il vendutissimo *Xenon II* della Mirrorsoft e *R-Type II*, un gioco a gettone non ancora arrivato in Italia ma che è diventato una vera mania in Giappone e in qualche altro Paese. Entrambi i programmi presentano lunghi alienicidi di massa, raffigurati con una fantasia grafica che arriva quasi a meritare la definizione di arte computerizzata.

Ma veniamo a *X-Out*: il gioco ha un'ambientazione subacquea, ma questa è l'unica informazione che siamo in grado di dare sullo scenario e sui retroscena. Gli autori hanno pensato bene di non fornire nessuna trama, limitandosi a inserire nella confezione un paio di paginette fotocopiate con un elenco dei comandi da usare e le istruzioni di caricamento. Per quanto un background non sia effettivamente indispensabile, va notato quanto sia strana questa scelta, in un periodo in cui i cosiddetti "scenari" vengono forniti persino nelle sale giochi.

La meccanica di gioco è alquanto bizzarra: dopo la lunga e notevole sequenza introduttiva, presa di peso dai vari "manga" giapponesi, ci viene fornito infatti un credito con il quale dobbiamo acquistare il nostro mezzo di combattimento. La scelta è fra quattro tipi di navicelle, diverse sia per l'aspetto esteriore che per l'armamento. Il micragnoso alieno dal quale acquistiamo il nostro arsenale ci impone di pagare una per una le armi montate, che possiamo selezionare da un menu iconico che offre addirittura 26 possibilità, variamente combinabili.

La scelta delle armi è tra le migliori mai viste in programmi di questo tipo: le opzioni comprendono tre tipi di laser moltiplicati per tre potenze differenti, tre modelli di missili a ricerca automatica, una smart bomb, lanciafiamme, laser binari, convergenti e "Balkan", scudi e armi orbitanti in grado di seguire sette diverse traiettorie. Naturalmente ogni arma ha un prezzo, e si deve scegliere fra una sola navicella armata all'inverosimile o una squadriglia di mezzi (sino a quattro) più deboli ma che offrono maggiori possibilità di resistere al fuoco nemico. Fuoco che, peraltro, è preciso e abbondante. Durante l'attraversamento dei livelli a scroll orizzontale, lo schermo si satura di proiettili, missili, bombe, mine, alieni kamikaze e ogni altra diavoleria, costringendoci a manovre ai limiti dell'impossibile, in grado di mandare in crisi anche i giocatori più esperti. Il nostro mezzo può muoversi a quattro velocità, premendo gli appositi tasti, e una delle prime cose da imparare in *X-Out* è proprio il modo di sfruttare al meglio le possibilità di movimento.

Durante il combattimento si respira a pieni polmoni l'atmosfera di *R-Type* (recensito su *Commodore Gazette* numero 2/89). Il nostro compito è quello di eliminare miriadi di creature impiegando i laser di bordo o le armi secondarie che si attivano tenendo premuto il pulsante di fuoco.

I livelli sono divisi in due parti, fra le quali lo scroll si ferma per permetterci di combattere una creatura più grossa delle altre. Nel primo livello troviamo un tizio armato di lanciamissili che si sposta sul dorso di un serpente di mare. La seconda pausa avviene in prossimità della fine del livello, quando incontriamo l'immancabile potentissimo "guardiano". In questi casi, il trucco sta nel trovare il suo punto debole che, colpito due o tre miliardi di volte, lo farà passare a miglior vita. Ovviamente i guardiani non sono molto desiderosi di questo cambiamento, e si difendono con ogni arma concepibile, non disdegnando di assalire il nostro veicolo di persona, smontandosi, inseguendoci, estroflettendo tentacoli e violando palesemente ogni convenzione e trattato di guerra.

Se dalla nostra descrizione *X-Out* vi è sembrato un gioco come tanti altri, ci affrettiamo a dire che vederlo sul monitor dell'Amiga non dà affatto quest'impressione. La quantità di oggetti in movimento è enorme, e l'accuratezza del disegno degli sprite supera persino quella di alcuni coin-op di recente produzione. Sulla confezione del programma viene osannata una frequenza di schermo di 50 fotogrammi al secondo, e in effetti la fluidità dell'azione è eccellente. Sul piano grafico, muovere delle critiche a

X-Out è impossibile: clichè a parte, ogni elemento è stato disegnato, ombreggiato e animato in maniera superba.

Pollici alzati anche per l'audio: l'onnipresente Chris Huelsback ha creato brevi brani d'atmosfera che costituiscono un ottimo complemento della sezione grafica.

Quando prendiamo in considerazione la giocabilità di questo mostro di programmazione, però, cominciano le dolenti note: *X-Out* è difficile. Molto difficile. Riuscire a superare il primo livello ci ha portato via una giornata intera, solo per rivelare un quadro ancora più denso di pericoli e di nemici. Se all'inizio questa può sembrare una caratteristica positiva, bisogna ricordare che a differenza di altri programmi qui si è praticamente obbligati a possedere una sola navicella, che una volta esaurita l'energia viene eliminata. Considerato che, dopo ogni decesso, per tornare all'inizio del primo quadro bisogna sopportare un lungo caricamento, l'intera procedura di acquisto della navicella e un secondo caricamento, il meccanismo diventa presto insopportabile.

Il secondo problema, di ordine etico più che pratico, sta nel valore di un programma clonato: non ci sono dubbi che il risultato sia spettacolare, ma dopo migliaia di critiche alla mancanza d'inventiva dei programmatori, non è forse il caso di "spingere" i prodotti originali comprando il vero *Xenon II* o il vero *R-Type II*, di cui la conversione per home computer non tarderà ad arrivare?

Chiudiamo ricordando che di *X-Out* esiste anche una versione per il buon vecchio Commodore 64, che purtroppo non abbiamo ancora avuto modo di provare, ma che si preannuncia tecnicamente ancor più spettacolare della controparte per Amiga, di cui dovrebbe mantenere ogni caratteristica.

F.R.

F-16 COMBAT PILOT

Falcon viene abbattuto dal nostro nuovo simulatore di volo!

Computer: C-64/128/Amiga
Versione: Amiga
Supporto: Disco
Prezzo: L. 59.000
Produzione: Digital Integration
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

«Quello che stai per affrontare è molto più di un videogioco, è il più realistico approccio possibile alla realtà del volo da caccia». Queste parole contenute nella premessa del manuale (96 pagine in italiano!) sono per una volta qualcosa di più di una semplice pubblicità. Ci troviamo infatti davanti al più completo e ben realizzato simulatore di volo oggi disponibile per l'Amiga.

La nostra carriera di piloti militari ha inizio in uno squadrone operativo in zona di guerra. Arrivati alla base è bene per prima cosa scrivere il proprio nome sulla scheda personale in modo che tutte le nostre esperienze di volo vi vengano registrate.

Si passa quindi alla schermata di selezione delle missioni. Cinque sezioni di un pentagono rappresentano ciascuna un ruolo nel quale può essere impiegato l'F-16: intercettazione aria-aria di aerei nemici; operazione offensiva contro un obiettivo al suolo (aeroporto, base militare, centro informativo, centro radar, postazione SAM o di artiglieria); missione d'interdizione contro depositi di benzina, centrali elettriche o fabbriche; supporto tattico sul campo di battaglia (distruzione di mezzi corazzati); ricognizione su postazioni nemiche (bisogna trasmettere alla propria base i dati rilevati).

Una volta completate tutte e cinque le missioni, si può accedere alla missione rappresentata al centro del pentagono: "Conquest" (conquista). Si tratta di fronteggiare l'attacco del nemico in qualità di comandante di squadrone. Non siamo più semplici piloti che devono eseguire gli ordini dei superiori, ma siamo noi stessi a scegliere la missione, valutando di persona quale sia la più adatta per prevenire le mosse del nemico o colpirne i punti più deboli. A questo punto impersoniamo il pilota che deve portare a compimento l'azione, e se veniamo abbattuti passiamo a impersonarne un altro. Dopo aver completato con pieno successo una campagna, ci viene offerto il passaggio a uno squadrone di più alto livello.

La selezione dell'armamento (sinistra) e la visuale destra dal posto di pilotaggio (destra). Gli indicatori accesi segnalano avarie a numerosi sistemi: siamo stati colpiti!

La schermata di selezione delle missioni presenta anche altre due possibilità. "Training" permette di addestrarsi all'atterraggio, alle

manovre di volo, alle intercettazioni e così via. "Gladiator" offre invece l'opportunità di collegare il proprio computer a quello di un amico attraverso un cavo RS232 per cimentarsi in entusiasmanti duelli aerei fino all'ultimissimo missile.

Dopo aver effettuato la scelta relativa alla missione, si passa allo schermo elettronico strategico, una vera e propria carta geografica digitale sulla quale, oltre all'aeroporto del giocatore, sono visualizzate tutte le installazioni militari amiche e nemiche. Non resta che vedere quale missione ci è stata assegnata (oppure assegnarcela di persona, se si tratta dell'operazione "Conquest") e inserire nel computer di navigazione dell'F-16 il nostro piano di volo. Da questo momento in poi, un apposito quadro ci terrà informati durante il volo sulla rotta da seguire per raggiungere i nostri obiettivi, e non rischieremo di perderci nei cieli. In questa fase si può anche scegliere se modificare le condizioni meteorologiche impostate dal programma, intervenendo su venti, turbolenze e nubi. Si tratta di una caratteristica davvero apprezzabile, soprattutto se si considera che si può scegliere anche tra il giorno e la notte.

L'ultima importante formalità prima del decollo è la selezione dell'armamento del proprio caccia. Visto che ritrovarsi a dover fronteggiare il nemico senza le armi adatte può essere decisamente spiacevole, è bene scegliere con molta attenzione tra i missili aria-aria Sidewinder e Amraam, quelli aria-suolo Harm antiradar e Maverick a guida infrarossa e laser, le bombe Durandal e Mk82-3-4; si decide infine se portare con sé serbatoi esterni, sistemi di ricognizione e di puntamento. Le scelte possono essere effettuate direttamente dal giocatore oppure si può far ricorso all'esperienza del personale di terra, e il tutto avviene in una schermata molto ben realizzata e anche decisamente simpatica (il pilota muove la testa seguendo il mouse...).

Siamo in fase di atterraggio, a 63 piedi di altezza sulla pista. Da notarsi la strumentazione e in particolare i tre schermi multifunzione nella parte inferiore

Finalmente siamo ai comandi dell'F-16! La prima impressione all'apparire del cruscotto dell'aereo è di un pannello veramente ben disegnato e contemporaneamente molto complicato. Non siamo infatti davanti a uno spara-e-fuggi qualunque, bensì al più dettagliato e realistico simulatore di volo da caccia oggi disponibile per l'Amiga. Sul quadro trovano posto molti strumenti, vediamoli con ordine.

Una piacevole novità rispetto ad altri prodotti di questo genere è costituita dalla presenza di tre ampi schermi multifunzione, ognuno dei quali può essere utilizzato, in base alle scelte del giocatore, come radar aria-aria per il tracciamento e l'inseguimento degli aerei nemici, radar aria-terra per la ricerca degli obiettivi al suolo, mappa digitale, indicatore dei sistemi d'arma, orizzonte artificiale digitale, indicatore di volo (velocità, quota, velocità verticale, direzione e carburante), schermo termico (visualizza un'immagine in miniatura dell'obiettivo), indicatore per l'atterraggio strumentale (ILS). Questa scelta degli schermi multifunzione è decisamente pratica ed è un'ottima soluzione ai limiti imposti dalle dimensioni del monitor del computer, non certo all'altezza del pannello di un vero F-16. Di un certo effetto sono in particolare l'orizzonte digitale e lo schermo termico.

C'è inoltre un ricevitore che funziona rilevando le onde radar emesse dal nemico, l'orizzonte artificiale, gli indicatori dell'angolo di attacco, della velocità verticale, dei giri del motore, dell'inserimento del postbruciatore, del carburante. Sul pannello di controllo frontale possono essere visualizzati i dati di distanza, direzione e tempo stimato d'arrivo relativi agli aeroporti e agli obiettivi (per gli obiettivi aerei il tempo d'arrivo è sostituito dalla quota). Sulla parte sinistra dello schermo sono collocate sei spie relative al funzionamento della strumentazione da ricognizione, del tracciamento dei bersagli, del radar, dell'ILS, del pilota automatico e del trasmettitore UHF. Il quadro di ricezione UHF serve

essenzialmente per ricevere messaggi come per esempio le istruzioni per l'atterraggio (direzione da seguire, quota, correzioni di rotta...). Infatti, se si contatta la torre prima del decollo si riceve l'autorizzazione a occupare la pista.

Non mancano gli indicatori relativi agli aerofreni, ai freni delle ruote, allo svuotamento dei serbatoi supplementari, all'esaurimento del carburante, al carrello, allo sgancio di serbatoi o armi, all'allarme per avarie, al blocco del reattore e all'incendio a bordo. Il tutto è completato dal pannello degli allarmi con spie per la rilevazione di missili terra-aria e aria-aria, di contromisure elettroniche da parte del nemico, di aggancio dai sistemi radar nemici sia al suolo che aviotrasportati.

Se a quanto abbiamo visto fino a questo momento uniamo la presenza dello HUD (uno schermo proiettato davanti alla visuale del pilota) con dati sul velivolo come velocità, quota, inclinazione, G, Mach, direzione e arma attiva, e i reticoli che permettono di allinearsi con l'obiettivo e fare fuoco al momento opportuno, possiamo affermare senza ombra di dubbio che la strumentazione è davvero completa. Ma non c'è da spaventarsi: gli strumenti sono tanti, ma anche facili e intuitivi da usare. In breve tempo, infatti, si acquisisce una piena padronanza del mezzo e si è pronti per sfidare il nemico.

Per quel che riguarda le visuali, è stata fatta una scelta perfettamente in tono con un simulatore degno di questo nome. Avete mai sentito parlare su un vero aereo di un'opzione di volo che permetta al pilota di premere un pulsante e vedere il suo aereo dalla torre di controllo o da un altro aereo? Certo che no. E così si è evitato di sprecare memoria preziosa per inserire le viste esterne che magari sono divertenti, d'effetto, ma nulla hanno della simulazione. Tutti i byte risparmiati sono stati consacrati a migliorare l'accuratezza del prodotto finale. E quindi non c'è da stupirsi se rullando verso l'hangar si esce dalla pista e il rumore dell'aereo si fa più intenso per il terreno più irregolare...

Non mancano invece le visuali dalla cabina. Quella posteriore, quella laterale sinistra dov'è collocata la mano del pilota sulla manetta che... udite, udite, per la prima volta nella storia dei simulatori per l'Amiga si muove seguendo la spinta imposta dal giocatore (ottimo!), e infine quella laterale destra dove oltre alle spie d'allarme relative ai malfunzionamenti dei vari apparati è collocata una bussola (qui, invece, la mano sulla cloche non si muove; peccato).

La visualizzazione tridimensionale del paesaggio circostante è di ottimo livello e la risposta ai comandi è decisamente buona. Sicuramente positivo è anche il voto da assegnare alla fluidità e alla rapidità delle animazioni. Chi scrive ha gradito in modo particolare il volo notturno, che è suggestivo e realistico, così come la presenza del pilota automatico per un atterraggio totalmente strumentale (ma non mancano le istruzioni dalla torre di controllo e l'ILS per un atterraggio effettuato dal giocatore seguendo le direttive del controllore di volo).

La giocabilità di *F-16* non è forse quella di *Interceptor*, ma il suo punto di forza è il realismo, ed è inevitabile che questo metta un po' in ombra un elemento come la giocabilità. Le missioni sono comunque avvincenti e cariche di tensione.

Ci troviamo sopra un obiettivo al suolo. Rispetto alla foto della pagina precedente è stata modificata la configurazione dei tre schermi multifunzione in "attacco al suolo"

Insomma, *F-16 Combat Pilot* ci è piaciuto moltissimo e gli conferiamo senza esitazione la palma del miglior simulatore di volo per l'Amiga, togliendola a *Falcon* che l'ha degnamente conservata sino a oggi.

Un ultimo avvertimento, prima che vi lanciate nel vostro primo volo sull'F-16: come nella realtà, se si compiono manovre troppo ardite per il fisico umano, si perdono i sensi per alcuni istanti (lo schermo si oscura) e non ci si può nemmeno lanciare col paracadute. Fate molta attenzione, non vorremmo avervi sulla coscienza.

M.L.

SPACE ROGUE

Poligoni pieni, gioco di ruolo e perfezione tecnica, nel miglior programma da due anni a questa parte

Computer: C-64/128
Supporto: Disco
Prezzo: L. 21.000
Produzione: Origin
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Questa volta non diremo niente della Origin. Chi legge *Commodore Gazette* conosce bene questa software house americana e l'altissima qualità dei suoi prodotti.

Potremmo fare un discorso introduttivo sulla grafica a poligoni pieni e sulle relative tendenze di mercato, ma dovremmo ripetere quanto già detto sul numero scorso di questa rivista. Non ci resta quindi che passare direttamente alla trama di *Space Rogue*, rimandandovi comunque alle precedenti recensioni di giochi appartenenti a questo filone per una panoramica sugli argomenti a cui abbiamo accennato.

Il gioco è ambientato in un lontano futuro, nella Nebulosa del Braccio Esterno di una galassia non meglio identificata: il nostro alter ego digitale veste la tuta spaziale di un cadetto, trasformato suo malgrado in pilota solitario in lotta per la salvezza dell'universo. Secondo quanto viene narrato nel breve racconto introduttivo (intitolato *Stelle di Opportunità*), il nostro personaggio sembra essere un replicante di Luke Skywalker, l'eroe di *Guerre stellari*. I suoi genitori, infatti, sono ricchi mercanti di mentalità piuttosto ristretta, che considerano quantomeno malsana la passione per lo spazio e per la navigazione astrale.

Dopo un lungo addestramento, nel corso del quale impara tutto quello che c'è da sapere sulla navigazione stellare, il nostro personaggio si imbarca sulla nave spaziale *Princess Blue*, e inizia una vita piuttosto tranquilla fra un balzo spaziale e l'altro. Un giorno, attraversando una zona teoricamente priva di pericoli, la *Princess Blue* capta un segnale di soccorso proveniente da una piccola nave di classe Sunracer, apparentemente abbandonata dal pilota. Mandato in missione esplorativa sulla *Jolly Roger* (questo il nome della nave alla deriva), l'eroe della vicenda vede l'equipaggio della propria nave sterminato da un attacco a sorpresa degli alieni Manchi, che non si curano del minuscolo Sunracer. Rimasto solo, legittimo proprietario di una nave piccola ma efficiente, in una galassia prossima a essere conquistata da una razza di alieni insettoidi, il protagonista viene affidato all'intraprendenza e all'astuzia del giocatore.

Come già accennato, lo scenario di *Space Rogue* è quello del Braccio Esterno, una sorta di confine fra la zona controllata dall'Impero (i buoni) e quella popolata dai violentissimi e insondabili Manchi. Gran parte del gioco vede il protagonista in viaggio fra gli otto sistemi della Nebulosa, uniti fra loro da particolari distorsioni spaziotemporali chiamate "Cancelli Malir". Nella confezione troviamo una mappa discretamente dettagliata che fornisce tutti i dati astronomici relativi alla navigazione fra un cancello e l'altro.

Trattandosi per lo più di un gioco di pilotaggio, lo schermo principale di *Space Rogue* si può considerare quello raffigurante la strumentazione del nostro Sunracer, che comprende un grosso schermo sul quale appare un'elaborazione computerizzata della situazione esterna. Pilotare il Sunracer è relativamente semplice, grazie anche al dettagliatissimo manuale di guida incluso nella confezione, del tutto simile a quello che si trova a corredo di una qualsiasi automobile. La cosa che più impressiona in questa sezione è senza dubbio il tipo di grafica impiegata: a poligoni pieni, completamente colorata, non ha nulla da invidiare a quella di alcuni ottimi prodotti per l'Amiga. Fatte le debite proporzioni, possiamo paragonarla tranquillamente a quella di *Starglider 2* per rapidità d'animazione e numero di oggetti in movimento; la fluidità non è naturalmente la stessa, ma sono del tutto assenti gli "scatti" cui ci avevano abituato i programmi di questo genere pubblicati sinora.

I movimenti del nostro veicolo avvengono su tre diverse scale: sistemi, settori e locale. La navigazione fra i sistemi prevede l'impiego dei Cancelli Malir, strane distorsioni spaziali che possiedono effettivamente la forma di un tunnel. I Cancelli sono stati costruiti da una razza aliena ormai scomparsa e, per quanto siano utilissimi, racchiudono grandi pericoli. Per superarli con successo è necessario imboccarli ad alta velocità, seguendo poi uno spettacolare e strettissimo "corridoio di volo" che ricorda il vecchio *Master of the Lamp* della Activision: uscire dal corridoio significa tornare al punto di partenza, mentre rallentare troppo provoca un rapido quanto pericoloso danneggiamento dello scafo.

Una volta uscito da un Cancello, il Sunracer si ritrova in un nuovo sistema planetario. A questo punto è possibile attivare il computer di bordo, che ci presenta una mappa elettronica del sistema suddivisa in settori. Per viaggiare da un settore all'altro è sufficiente spostare un cursore, e il calcolatore si occupa automaticamente del viaggio.

Arrivati nel settore di destinazione, entra in gioco il terzo tipo di navigazione, gestito nuovamente tramite lo schermo princi-

pale: qui è possibile selezionare con un sistema computerizzato il nostro bersaglio e prenderlo di mira con precisione assoluta. Le nostre intenzioni non devono essere necessariamente bellicose; all'interno di un settore, infatti, è possibile trovare una gran varietà di oggetti, che vanno da enormi soli incandescenti a basi imperiali, stazioni minerarie, piattaforme indipendenti e persino una colossale portaerei.

servata sotto tre prospettive differenti, di cui due dall'esterno della nave.

Un'ultima osservazione sulle tecniche di pilotaggio: il Sunracer può ruotare contemporaneamente su tre assi e utilizzare due sistemi di guida, di cui uno "standard" e uno basato sull'impiego di vettori, particolarmente utile quando si vogliono raggiungere velocità elevate, ma terribilmente complicato; si deve infatti tener

Prima di raggiungere queste mete, purtroppo, bisogna affrontare numerosi ostacoli, costituiti da asteroidi, cristalli di antimateria esplosivi o le più prosaiche ma non meno pericolose astronavi avversarie.

Space Rogue non pone particolare enfasi sul combattimento, tuttavia offre un'impressionante varietà di armi, difese e tecniche di battaglia. Nelle sequenze di scontro con altre astronavi si può apprezzare appieno l'eccellenza delle routine grafiche, che non subiscono rallentamenti sensibili nemmeno quando gestiscono contemporaneamente quattro astronavi, una decina dei loro missili e innumerevoli asteroidi. L'azione, inoltre, può essere osconto dalla presenza dei campi gravitazionali di tutti gli oggetti che compaiono sullo schermo.

Quando in un modo o nell'altro riusciamo ad agganciarci a una qualsiasi struttura, ecco che *Space Rogue* rivela la sua seconda natura di gioco di ruolo (!). La visuale cambia per mostrarci uno schermo scorrevole simile a quelli di *Ultima* (non per niente il gioco è della Origin), nel quale il nostro personaggio può esplorare il luogo in cui è giunto, interagendo con i suoi abitanti. Durante le prime fasi del gioco, la funzione principale di queste sezioni è di fornire un luogo in cui riparare i danni della nave, ma col procedere della partita vengono alla luce possibilità sempre nuove e inaspettate.

Viaggiando fra una base e l'altra, per esempio, si può diventare ricchi tramite il commercio di merci preziose. Chi ha giocato a *Elite* troverà pane per i suoi denti, considerata la severità dei controlli eseguiti dall'Impero su alcune merci e le complesse leggi commerciali che regolano questo tipo di attività. Inoltre, parlando con gli abitanti dei luoghi (o strutture) che visitiamo, spesso si aprono nuove opportunità: c'è chi ha bisogno di piccoli favori interplanetari, chi cerca la Verità Universale, chi ha intenzione di sovvertire l'Impero e chi ce l'ha con la Gilda dei pirati, che peraltro sono prontissimi ad accogliere un nuovo confratello. Gli incontri sono resi ancor più godibili dalla presenza di personaggi perfettamente caratterizzati, che spesso ricalcano pesantemente quelli della trilogia di *Guerre stellari*. Frequentandoli, scoprirete le caratteristiche di numerose razze aliene, e alla fine vi troverete invischiati in un conflitto intergalattico di cui risulterete unico arbitro. Piuttosto stimolante, non è vero?

Potremmo parlare di *Space Rogue* ancora per pagine e pagine, descrivendone in dettaglio le numerosissime caratteristiche ed elencandone i "tocchi di classe" che lo distinguono dagli altri giochi di questo genere (che comunque non sono molti, purtroppo). La cosa migliore, in ogni caso, è consigliarvelo senza riserve. *Space Rogue* è senza dubbio il programma più completo e giocabile che sia stato pubblicato da un paio d'anni a questa parte, e piacerà sia agli amanti dell'azione pura che agli appassionati dei giochi di ruolo. Sfortunatamente, la versione Amiga uscirà tra parecchi mesi, e prevediamo che nel frattempo ci saranno parecchi possessori del 16 bit che invidieranno i "piccoli" C-64.

F.R.

THE HOUND OF SHADOW

Un debutto poco felice per il sistema "Timeline"

Computer: Amiga
Supporto: Disco
Prezzo: L. 45.000
Produzione: Electronic Arts
Distribuzione: C.T.O. (Via Piemonte 7/F, 40069 Zola Predosa - 051/753133)

Non si può certo dire che ai progettisti della Electronic Arts manchi la fantasia: sono stati gli inventori dei giochi con editor (*Pinball Construction Set*) quando tutti gli altri programmi in circolazione costringevano gli utenti a seguire rigidi schemi, e sono stati sempre loro a scatenare il boom dei giochi di ruolo (*Wasteland* e la serie di *Bard's Tale*) mentre il resto dei produttori sfornava semplici adventure. Insomma, i programmatori della EA hanno sempre cercato di restare un passo avanti al gruppo.

L'ultima intuizione dello staff americano riguarda il mondo ormai vastissimo dei giochi di ruolo per computer: osservando l'incredibile raffinatezza raggiunta dalle interfacce grafiche e di comando di questo genere di programmi (si veda la recensione di *Knights of Legend* su questo stesso numero), qualcuno alla Electronic Arts ha pensato che fosse il momento di cambiare rotta.

L'idea che sta alla base di *The Hound of Shadow* è di riportare il role playing game alla sua originale forma d'intricato racconto, nel quale saltuarie schermate grafiche si affiancano a migliaia e migliaia di parole, capaci di creare quell'atmosfera da romanzo assolutamente impossibile da trovare persino nel miglior prodotto grafico. Il programma segue quindi lo stile delle "vecchie" e indimenticabili avventure della Infocom, con lunghe descrizioni testuali nelle quali ogni parola è calibrata per evocare un mondo fantastico plausibile e dettagliato.

L'interfaccia utente è affidata a un parser tradizionale, capace d'interpretare comandi scritti in lingua inglese.

Sempre seguendo lo spirito dei giochi di ruolo giocati con carta, penna e qualche dado, *The Hound of Shadow* permette all'utente di controllare solo il protagonista della vicenda, lasciando il computer alla guida di tutti gli altri personaggi.

E a proposito di personaggi, eccoci giunti alla fase di generazione del protagonista dell'avventura, che si effettua tramite un inusuale sistema di cursori e grafici a barre. I personaggi di *The Hound of Shadow* hanno a disposizione un certo numero di "punti-abilità" che ne qualificano il campo di specializzazione: c'è il personaggio fisicamente possente, capace di vere prodezze acrobatiche e abilissimo nell'uso delle armi, come c'è il "tranquillo" studioso che non vincerà mai il titolo di Mister Universo ma che se la cava meglio del primo con strumenti scientifici e antichi linguaggi da decifrare.

Una serie di sotto-menu ci permette di assegnare i punti-abilità in maniera ancor più dettagliata, creando per esempio ottimi cecchini totalmente incapaci di destreggiarsi con la spada, oppure occultisti specializzati nei riti celtici ma del tutto ignoranti in quelli delle culture tribali africane. Una volta terminata questa procedura (non particolarmente complicata, comunque) gli ultimi dati da inserire riguardano l'età del personaggio, il sesso, la nazionalità (inglese o americana) e il ruolo ricoperto durante la Prima guerra mondiale (la storia è ambientata nei primi anni Venti). E qui entra in campo il "Timeline".

Presentato dai suoi autori come una rivoluzione assoluta nel campo dei giochi per computer, il "Sistema Timeline" è la routine attorno alla quale è stato costruito tutto *The Hound of Shadow*. Questo codice serve a creare all'interno del file relativo al nostro personaggio una serie di dati riguardanti la sua esistenza precedente: ciò significa che un ragazzino cresciuto per le strade non potrà mai possedere l'istruzione del figlio di un nobile britannico; che un soldato vissuto per tre anni in prima linea potrebbe portarsi dietro preoccupanti disturbi psichici; che un playboy americano, pur avendo molti difetti, avrà senz'altro viaggiato molto, sviluppando così una conoscenza delle culture straniere e una notevole abilità nelle relazioni sociali. Via via che il gioco procede, questi dati dovrebbero essere aggiornati, delineando alla fine un personaggio particolarmente dettagliato. Ma possono sorgere problemi: a noi, per esempio, è capitato un caso in cui, dopo essere stato assalito da uno sciame di mosche sul luogo di un orrendo delitto, il nostro personaggio ha sviluppato una sorta di fobia per cui il ronzio di un insetto gli evocava visioni angoscianti di persone smembrate e putrefatte.

No, non siamo impazziti, anche se molti si chiederanno che cosa hanno a che fare immagini simili con un gioco di ruolo.

In effetti, sinora non abbiamo avuto modo di parlare della trama del gioco, che costituisce già di per sé una notevole innovazione. *The Hound of Shadow* si basa sulle opere di Howard Phillips Lovecraft, uno scrittore neogotico americano, nato negli ultimi anni del secolo scorso, la cui fama è in lentissima ma costante crescita. Quindi chiariamo subito che in questa sede ci limiteremo a brevi cenni, della cui incompletezza ci scusiamo in anticipo con i molti fan del "solitario di Providence".

La caratteristica saliente di Lovecraft è di aver creato una

complessa teogonia ctonia che sostiene molti suoi racconti, fornendo loro una matrice comune. Sintetizzando al massimo la mitologia lovecraftiana, possiamo descrivere l'Universo come un infinito campo di battaglia sul quale hanno lottato e continuano a scontrarsi entità innominabili divise in fazioni che non sono identificabili come "Bene" o "Male".

In questo quadro, la razza umana ricopre un ruolo del tutto secondario, che la relega costantemente nella sconcertante funzione di vittima, a volte schiacciata senza che le entità superiori nemmeno si accorgano della sua esistenza. In effetti, visto che il pensiero di combattere simili mostri è inconcepibile, nella maggior parte delle storie di Lovecraft i protagonisti hanno a che fare con i servitori più infimi di questi esseri, che vengono identificati con le creature più tenebrose delle varie mitologie: vampiri, ghoul, demoni...

The Hound of Shadow racconta la vicenda di un "investigatore" (il giocatore) rimasto coinvolto in una diabolica vendetta pandimensionale e perseguitato da una crudelissima e inarrestabile creatura infernale, ovvero il mastino dell'ombra che dà il titolo al programma. Fra omicidi rituali, sedute spiritiche, ricerche in biblioteca e qualche momento di azione frenetica, il protagonista cerca di salvarsi dal suo orrendo destino, ma nel farlo incappa in qualcosa di ancora più terribile, che dovrà affrontare con le poche conoscenze a sua disposizione.

Ambientato nella Londra d'inizio secolo, ricreata con efficaci descrizioni e appropriate illustrazioni in color seppia, *The Hound of Shadow* potrebbe affascinare ben più di un giocatore: viste le idee di base e le tecniche impiegate era lecito aspettarsi un capolavoro.

Ma purtroppo le cose non stanno così. Forse a causa dell'eccessivo interesse per il Timeline e per la trama, gli autori si sono dimenticati che ci voleva anche un programma degno di questo nome. Una volta superate le prime fasi del gioco, infatti, ci si rende conto di trovarsi di fronte a un adventure estremamente primordiale, la cui unica caratteristica notevole è quella di adeguare descrizioni di oggetti e azioni alle abilità da noi selezionate nella creazione del personaggio.

In tutte le altre fasi, appare evidente che la struttura del programma è simile a quella di certe pessime avventure italiane, in cui il computer non faceva procedere la vicenda sino a quando non veniva inserito un comando ben preciso, a volte impossibile da indovinare; un esempio che vale per tutti è l'input necessario per raggiungere l'abitazione di un certo signor Talbot, abitante al 34b di Dean Street: vengono rifiutati GO TO TALBOT, GO TO DEAN STREET, GO TO 34, DEAN STREET e decine di comandi simili sino a che, nella disperazione più nera, non s'inserisce GO TO 34b OF DEAN STREET, unico input capace di ottenere l'effetto voluto. Poiché l'estrema sequenzialità della vicenda costringe il protagonista a compiere azioni ben precise (e spesso suggerite apertamente dal computer), potete ben immaginare lo sconforto del giocatore medio.

Concludiamo segnalando la presenza nella confezione di un utilissimo dizionario inglese-tedesco-francese-italiano, vera manna per i giocatori di adventure su computer, che tuttavia non riesce a far superare la delusione di chi aveva riposto tante speranze in un questo titolo. Lo consigliamo comunque agli adoratori di Lovecraft, e suggeriamo agli altri utenti di attendere i prossimi titoli della serie "Timeline", con la speranza che i programmatori non intendano perseverare (diabolicamente?) su questa strada.

F.R.

SPACE ACE

Gioco o cartone animato? Questo è il problema

Computer: Amiga
Supporto: Disco
Prezzo: L. 80.000
Produzione: ReadySoft
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)
Disponibile anche presso: Lago (Via Napoleona 16, 22100 Como - 031/300174)

Ciò che prevedevo non ha tardato a verificarsi: era abbastanza evidente, visto il discreto successo ottenuto da *Dragon's Lair*, che sarebbero state convertite per l'Amiga anche altre delle più note "laseravventure" da sala giochi. E in effetti è quasi sorprendente il grado di versatilità dimostrato da questo computer, che si dimostra pienamente in grado di gestire veri e propri cartoni animati. Nonostante la scarsa "interattività" fra gioco e giocatore, infatti, anche *Space Ace* si dimostra discretamente fedele alla versione su disco laser.

La grafica è sicuramente meno elaborata di quella messa in mostra da *Dragon's Lair*, ma le animazioni scorrono in maniera più rapida e più fluida e, soprattutto, il caricamento delle varie scene è molto più rapido. Per uno spettatore che guardi un abile giocatore, il risultato è la visione di un cartone animato molto ben disegnato, con una vicenda comprensibile e non troppo frammentaria.

Facciamo un po' di storia, a beneficio di chi non ha mai visto il gioco nelle sale: il terribile comandante alieno Borf sta attaccando la Terra servendosi del micidiale "Infanto Ray" un'arma che trasforma in innocui poppanti tutti coloro che vengono colpiti dal raggio. Solo il nostro eroe, Space Ace, è in grado di salvare il mondo, ma anche lui subisce gli effetti del malefico raggio (ed è per questo che lo vedete ragazzo, uomo e poi di nuovo ragazzo con il variare delle scene), e oltretutto Kimberly, la sua fidanzata, è stata rapita da Borf.

Ecco quindi gli obiettivi del gioco: penetrare nella fortezza di Borf, salvare Kimberly, e affrontare Borf in un duello all'ultimo sangue.

Il gioco è costituito da 33 scene in sequenza, che compaiono sempre nello stesso ordine; per superare una scena è necessario effettuare da una a quattro mosse nella giusta direzione e soprattutto al momento giusto (una frazione di secondo in più o in meno potrebbe costarvi la vita). Le mosse possibili sono solo cinque: le quattro direzioni più il pulsante Fire per usare le armi che ogni tanto vi ritrovate in mano (si può usare tanto la tastiera quanto il joystick). Dopo aver perso tre vite, il gioco finisce e bisogna ricominciare dalla prima scena. Terminate tutte e 33 le scene (quattro dischi), il gioco finisce e, da un certo punto di vista, non ha più senso giocare, in quanto si rigiocherebbe ogni volta una partita uguale alla precedente (vi raccomandiamo quindi di utilizzare la soluzione pubblicata in questo stesso numero con la dovuta cautela).

Proprio questa invariabilità degli schemi di gioco è fonte di accese discussioni in merito alla validità dei videogiochi "tipo-cartoon". Da una parte c'è chi sostiene che non siano nemmeno definibili "giochi", ma che si debbano descrivere come pure esibizioni di computergrafica fine a se stesse, graziose, ben realizzate, ma del tutto prive d'interesse una volta che si sia arrivati fino in fondo. La spesa, si dice, è decisamente eccessiva per una fonte di divertimento tanto limitata. Su posizioni opposte si schiera chi proclama la validità del gioco in virtù delle sue eccellenti doti grafiche, e la sua originalità rispetto alle tradizionali impostazioni dei videogiochi.

Personalmente non propendo né per l'una né per l'altra tesi. Non è una questione di diplomazia, ma in effetti ritengo che, fino alla sua soluzione, il gioco sia divertente, emozionante, piacevole da vedere e da sentire; dopo, sono d'accordo, diventa quasi inutile continuare a giocare partite dall'esito e dallo svolgimento

scontati. Del resto anche le omonime versioni da bar dei giochi, nonostante le maggiori capacità dei videodischi e la non sequenzialità delle scene, ripresentavano lo stesso problema anche se in modo meno evidente. E temo che gli appassionati arrivati alla soluzione finale di *Dragon's Lair* o di *Space Ace* al bar, una moneta dopo l'altra, abbiano speso una cifra che non si discosta troppo dal prezzo di vendita di questi programmi (una partita a *Space Ace* costava ai tempi 500 lire). Il problema non si pone in termini

di valutazioni oggettive, ma in termini di scelte soggettive. Questi giochi costituiscono un genere a sé stante (potremmo chiamarli "laseravventure") che, come tutti i generi, può piacere o non piacere: l'importante è che l'acquirente sia consapevole di quello che si troverà di fronte, e che non si lasci abbagliare dalle belle schermate pubblicate sul retro delle confezioni. È proprio in quest'ottica che conferisco un voto positivo al gioco, dal momento che nel suo genere non ha nulla da invidiare al tanto conclamato (o disprezzato) *Dragon's Lair*. **N.F.R.**

FIGHTER BOMBER

Un cacciabombardiere entra nel nostro spazio aereo digitale

Computer: C-64/128/Amiga
Supporto: Cassetta/Disco
Prezzo: L. 29.000/39.000/59.000
Produzione: Activision
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago - 0332/212255)

Dopo molte anticipazioni, finalmente *Fighter Bomber* ha raggiunto i computershop. Delusione o conferma di aspettative positive? Scopriamolo insieme.

Prima di tutto va messa in luce l'ambientazione del programma. In un panorama di distensione Est-Ovest come quello attuale, si è pensato bene di scegliere un obiettivo diverso dal solito nemico dalla stella rossa. Ci troviamo infatti alla base aerea statunitense di Ellworth, dove sta per svolgersi la "Gara annuale di bombardamento del comando aereo strategico", alla quale partecipano piloti militari americani, europei (nessun italiano) e anche russi. Il vincitore si porterà a casa il trofeo Curtis E. LeMay.

Nella prima schermata viene richiesto il nome del pilota, al quale verranno poi associati i vari esiti delle missioni. Da qui si passa a una caratteristica decisamente originale e apprezzabile: la selezione dell'aereo. Al contrario dei suoi concorrenti, questo gioco permette infatti di selezionare il proprio velivolo tra ben sette aeromobili: gli statunitensi F-15E Strike Eagle, F-4E Phantom e F-111F Aardvark, il Tornado IDS britannico e tedesco (manca solo quello con le insegne italiane!), lo svedese Saab AJ37 Viggen e infine il sovietico Mig-27. La versione per il C-64 riduce gli aerei a quattro escludendo F-15, Tornado tedesco e Saab. È importante sottolineare che non si tratta di una possibilità di scelta puramente simbolica; al contrario, a seconda dell'aeroplano selezionato variano le prestazioni, il carico di armamenti, il pannello comandi e il disegno dell'aereo che si vede selezionando le visuali esterne. Ma la varietà di aeromobili coinvolti non è ancora finita. La schermata successiva propone infatti un nutrito ventaglio di possibili aviogetti da assegnare all'avversario: gli statunitensi F-14 Tomcat, F-16 Falcon e F-5 Tiger, i sovietici Mig-29 Fulcrum, Su-27 Flanker e Mig-31 Foxhound, e

Sopra: siamo al posto di pilotaggio nella versione per C-64. Sotto: in fase di atterraggio nella versione per Amiga

infine il francese Mirage 2000. Ancora una volta la versione per il C-64 impone alcune limitazioni rispetto a quella per l'Amiga, riducendo il numero dei velivoli a tre ed escludendo F-16, Su-27, Mig-31 e Mirage 2000.

La selezione del livello offre due alternative principali. Con Free flight (volo libero) ci si può familiarizzare con i comandi e acquisire padronanza del velivolo sperimentando sette situazioni diverse: rullaggio, decollo, manovre in quota, bombardamento, volo sulla città, rifornimento in volo e atterraggio (con il C-64 si

può scegliere solo tra le prime quattro). Con la selezione delle missioni si inizia invece la qualificazione vera e propria, durante la quale tutto verrà registrato nella scheda personale del pilota. Le missioni sono sedici (otto per il C-64) divise in quattro gruppi dai

Rispetto alla foto precedente abbiamo cambiato aereo ed è cambiato il cruscotto (Amiga)

nomi suggestivi: missioni segrete, tattiche, strategiche e d'attacco. Le missioni sono disponibili soltanto in sequenza: per passare alla seconda è infatti necessario aver completato la prima e così via.

Vista esterna aerea (versione per l'Amiga)

Quando si seleziona una missione appare una schermata che ci dà le informazioni necessarie per portarla a termine, e poi un'altra per il carico dell'armamento sul jet; oltre al cannoncino già presente sul velivolo, si può scegliere tra un assortimento che prevede due tipi di missili aria-aria, due di missili aria-terra, tre di bombe, due dispensatori di sottomunizioni e un distributore di esche elettroniche (la versione C-64 è più limitata). Il munizionamento può essere scelto dal giocatore oppure caricato in modo automatico.

Sediamoci finalmente sul seggiolino del pilota. Nella versione per l'Amiga ci troviamo dinanzi a un pannello comandi ben disegnato ed essenziale, sul quale trovano posto: radar con relativo indicatore di portata, altimetro, bussola, orizzonte artificiale e indicatori relativi a posizione della manetta, carburante, stallo, aerofreni, freni, carrelli e incendio a bordo. Tre schermi forniscono informazioni relative alla vicinanza di nemici, alle armi selezionate, al bersaglio e alla rotta da seguire per raggiungere l'aeroporto o l'obiettivo. Lo HUD (uno schermo proiettato sulla visuale del pilota) offre invece con immediatezza i dati relativi a velocità, quota e direzione.

A questo punto va fatta una considerazione. Il miglior software di simulazione del volo pubblicato fino a oggi per l'Amiga ha proposto due scuole: da una parte *Interceptor* che offre pochi comandi, e quindi una minor fedeltà di simulazione, ma è dotato di un'elevatissima giocabilità con una strutturazione "a missioni" davvero appassionante; dall'altra *Falcon* che propone un pannello comandi più complesso, un grado di simulazione più vicino alla realtà, ma una giocabilità inferiore. *Bomber* è senza dubbio discepolo della prima scuola e dal momento che molti byte sono passati nei nostri disk drive dalla pubblicazione del suo programma-ispiratore, offre di più: la grafica è di livello decisamente superiore. Si concentra meno sui dettagli di simulazione e punta molto sulle visuali, che abbondano; citiamo tra l'altro le visuali dall'abitacolo (frontale, frontale estesa, da destra, da sinistra e posteriore), dal proiettile sparato, dall'aereo nemico, dalla torre di controllo, dal satellite, e infine la visuale "al seguito" e dall'esterno dell'aereo (con zoom e panoramica destra/sinistra, alto/basso). I risultati sono apprezzabili.

Gli aerei sono disegnati in modo preciso, con un buon effetto tridimensionale, e lo stesso vale per i particolari esterni. Si può dire che il risultato grafico è superiore non solo a *Interceptor*, ma anche a *Falcon*. Soddisfacente anche la fluidità dell'animazione.

Abbiamo lodato la grafica e l'animazione, ma la simulazione? Come abbiamo già detto, *Bomber* non appartiene alla scuola dei simulatori dettagliati, ma non si può nemmeno dire che sia un puro gioco di volo. Anche se molte delle finezze che fanno la felicità dei puristi della simulazione non sono incluse, il controllo che abbiamo sul velivolo e la buona risposta ai comandi è abbastanza interessante anche per chi non ama l'eccessiva semplicità.

Il combattimento aereo è piuttosto realistico, godibile e non

difficoltoso come l'attacco di bersagli al suolo. Utilissimo lo schermo che rileva la distanza e fornisce la direzione per raggiungere un bersaglio o la propria base. Con altri programmi del genere, infatti, spesso capitava di perdersi nell'immensità del blu.

L'obiettivo degli ideatori di *Bomber* è stato quello di offrire una grafica di alto livello con ottime visuali dall'esterno dell'aereo, una strumentazione semplice e poco complessa in modo da non scoraggiare gli amanti dei giochi di volo (ma non tanto semplificata da non essere di alcun interesse per chi ama le simulazioni accurate) e una serie di missioni alle quali accedere in scala progressiva (sullo stile di *Interceptor*) per dare spazio alla giocabilità. A tutto questo è stata aggiunta l'inedita possibilità di variare a piacimento gli aerei (amici e nemici) con conseguente mutazione della grafica del pannello strumenti e della visuale esterna dell'aereo.

I programmatori sono riusciti nel loro intento? Per quel che riguarda la grafica, pur non introducendo nulla di davvero eclatante, si può dire di sì. La semplicità dei comandi farà felici gli amanti dei giochi, decisamente meno quelli dei simulatori per i quali probabilmente *Bomber* si avvicinerà al livello di non gradimento... ma senza superarlo. I risultati sono senza dubbio peggiori nel campo della giocabilità. È buona, ma se andiamo a confrontarla – com'è d'obbligo, vista l'indubbia ispirazione al predecessore – con quella di *Interceptor*, dobbiamo riconoscere che il prodotto della EA ne esce vincente. La tensione e il desiderio di passare alla missione successiva suscitate da *Interceptor* qui non sono nemmeno sfiorate.

La versione per il C-64 è del tutto simile, a grandi linee, anche se le ovvie differenze tra le due macchine, impongono (oltre alle già citate limitazioni nelle scelte pre-gioco) anche l'eliminazione delle visuali frontale estesa, da destra, da sinistra e posteriore. La grafica viene sfruttata in modo da privilegiare fluidità e rapidità di controllo, piuttosto che gli abbellimenti esterni. Forse fin troppo: il paesaggio è praticamente sempre rappresentato da un interminabile fondo grigio! In questo formato, *Bomber* – privo della bella grafica e della ricchezza di particolari resa possibile dell'Amiga – risulta un puro e semplice gioco che non offre novità. Quello che nella versione per l'Amiga era un limite prossimo, qui è un limite valicato: difficilmente può interessare gli amanti della simulazione di volo accurata.

Siamo dinanzi a un ponte pronti per colpirlo (versione per il Commodore 64)

C'è un'opzione degna di particolare attenzione che permette all'utente di progettare personalmente le missioni. L'operazione è estremamente semplice. Su una carta geografica si selezionano la base, i bersagli (postazioni SAM, radar, convogli...) e gli eventuali punti di rifornimento in volo. A questo punto la missione è pronta per essere collaudata ed eventualmente modificata. Quando ci sembra soddisfacente, la schermata contenente la descrizione della missione può essere salvata su disco e, volendo, si può passarla a un amico per mettere alla prova la sua abilità di pilota. Quest'idea di progettare da soli le missioni è davvero nuova e intelligente, e merita pieno plauso, anche perché rende il gioco decisamente più interessante. Meno plauso merita il fatto che non sia disponibile nella versione C-64.

Riatterriamo ora sulla pista e veniamo alle nostre conclusioni. La versione per l'Amiga è in definitiva un buon prodotto, che ricalca appieno la filosofia di semplicità di casa Activision. Questo si traduce in un programma che può soddisfare molto gli amanti dei giochi di volo e un po' meno quelli delle simulazioni. Nel complesso un programma con molti punti d'interesse la cui non rigorosissima fedeltà di simulazione viene riscattata dalla possibilità di crearsi missioni personalizzate.

La versione per C-64 è un'altra cosa. Un gioco di volo (non un vero simulatore) senza peculiarità particolari e decisamente troppo inferiore alla versione Amiga. A quanto pare non hanno incluso neanche l'eiezione del seggiolino e io mi sto schiantando... È finita!

M.L.

PAGESTREAM

L'Amiga diventa una stazione di desktop publishing

Computer: Amiga
Supporto: Disco
Prezzo: L. 299.000
Produzione: Soft-Logik Publishing
Disponibile presso: Lago (Via Napoleona 16, 22100 Como - 031/300174)

Con *PageStream* è possibile scrivere, correggere, illustrare, e stampare documenti di alta qualità servendosi soltanto dell'Amiga. Il programma rappresenta una sintesi di fotocomposizione, disegno e word processor, che permette di ottenere col minimo sforzo il controllo più completo del documento.

Con un programma come *PageStream* si possono fare un'infinità di cose. È possibile creare manifestini, grafici, tabelle, lucidi da proiettare con la lavagna luminosa, opuscoli e biglietti da visita. È anche possibile realizzare veri e propri manuali e, perché no, anche una rivista. Il programma ci permette di utilizzare vari modi di visualizzazione e supporta una vasta gamma di stampanti, da quelle a matrice di punti fino allo standard Postscript.

Nella confezione si trovano un voluminoso manuale e alcuni dischi, il tutto racchiuso in un pratico cofanetto. Il manuale, in un raccoglitore ad anelli, è corredato da una serie di schede gialle che rappresentano un'alternativa all'indice analitico. La grafica è sobria e gradevole e i testi sono semplici ed esaurienti. Dopo una breve introduzione che ci fa entrare nel mondo dei termini grafici, vengono presentati tre brevi Tutorial che coprono tutte le principali caratteristiche del programma insegnandoci a utilizzare le funzioni di *PageStream*. Segue una sezione dedicata ai trucchi e alle scorciatoie che si possono usare in casi particolari, e quindi si passa alla guida di riferimento che spiega dettagliatamente, e con grande chiarezza, tutti i menu, le loro voci e le icone del "Toolbox". In fondo al manuale trovano posto l'elenco delle combinazioni di tasti tramite le quali si possono chiamare le funzioni dei menu, e l'elenco dei simboli speciali. Il manuale (in inglese) è completato da un glossario dei termini più significativi e da un indice analitico.

Il programma è distribuito in due o tre dischi (dipende dalla versione). Infatti sono attualmente in commercio sia la versione 1.6 che la 1.8. In quest'ultima dovrebbero essere scomparsi alcuni bug presenti nella 1.6. In realtà, nella nostra prova le due versioni ci sono sembrate assolutamente identiche.

Per valutare un programma, e in particolare un programma ricco di opzioni come questo, non c'è altra strada che provarlo a fondo. Da qui la decisione d'illustrare, almeno a grandi linee, una tipica sessione di lavoro con *PageStream*. Abbiamo immaginato di voler creare, manco a dirlo, una pagina di *Commodore Gazette*. Per semplicità, e per poter effettuare i dovuti raffronti, si è scelta una pagina pubblicata nello scorso numero della rivista. Naturalmente non possiamo aspettarci che il nostro lavoro risulti perfettamente identico all'originale perché non abbiamo a disposizione le stesse fonti-carattere della fotocompositrice della redazione. Quindi dovremo adattarci a quello che ci fornisce il programma. Comunque, se avrete la pazienza di seguirci, vi renderete conto che con poco sforzo si possono realizzare cose davvero interessanti.

Prima d'iniziare è bene vedere come si suddivide una schermata di *PageStream*, aiutandoci anche con le foto che accompagnano l'articolo. In alto, lo schermo è attraversato dalla barra dei menu che ospita i nove menu a discesa. Sulla destra c'è una sottile finestra che contiene varie icone: è il Toolbox. Grazie alle icone si possono inserire testi, selezionare oggetti, creare colonne, cambiare le dimensioni degli oggetti, disegnare varie figure geometriche, e ci si può muovere attraverso le pagine del documento. La restante parte dello schermo è l'area di lavoro dove si creano i documenti. È importante notare che *PageStream* è in grado di visualizzare più di un documento alla volta, ognuno in una finestra separata.

Iniziamo a lavorare. Per default il programma utilizza i pollici mentre noi scegliamo, tra le varie possibilità, i centimetri, un'unità di misura a noi più congeniale (possiamo anche salvare questo cambiamento per averlo sempre disponibile all'atto dell'accensione). A questo punto creiamo un nuovo documento usando il comando NEW del menu FILE. Un requester ci permette di scegliere le dimensioni e il tipo di documento che stiamo generando tra una vasta gamma di scelte preimpostate. Le varie possibilità che ci vengono offerte non si adattano al lavoro che intendiamo svolgere. Decidiamo quindi d'inserire manualmente le misure della pagina campione. Nell'apposito spazio introduciamo le misure 20,5 x 27,6 cm. Scegliamo quindi un documento a doppia pagina. A questo punto si apre una grande finestra: la nostra area di lavoro. Per default il programma ci mostra soltanto una parte del foglio di lavoro. Possiamo comunque spostarci in qualsiasi punto della pagina e cambiare come preferiamo la parte di foglio visualizzata, semplicemente scegliendo la voce adatta dal menu VIEW.

Prima d'iniziare a lavorare è opportuno avere ben chiaro il risultato che si vuole ottenere, per poter sfruttare appieno le opzioni che ci vengono messe a disposizione dal programma. Una rivista come *Commodore Gazette*,

graficamente è composta di alcune parti fisse e ben definite che permettono di dare omogeneità alla pubblicazione. Gli articoli, per esempio, sono scritti sempre con lo stesso tipo di carattere, i titoli sono sempre delle stesse dimensioni e sono giustificati a sinistra, e così via. Tutte queste parti fisse possono essere catalogate e archiviate per essere richiamate in seguito automaticamente. A questo scopo, *PageStream* fornisce i cosiddetti Tag. Un Tag non è altro che la descrizione di un elemento del documento.

La pagina che intendiamo ricostruire è la prima dell'articolo sulla computermusica apparso nello scorso numero. Trattandosi della prima pagina di un articolo presenta alcuni elementi particolari e un'impaginazione leggermente diversa da quella delle altre pagine della rivista. Proprio per questo motivo si presta bene per esaminare il modo di operare dei Tag. Guardando la pagina originale dall'alto verso il basso, incontriamo vari elementi: il primo è l'Occhiello, una breve frase che identifica l'argomento dell'articolo. Quindi c'è il Titolo, la Firma, il Sottotitolo e, finalmente, il testo vero e proprio. Tutti questi elementi hanno dimensioni e caratteristiche differenti che possono essere registrate come Tag. Partiamo dall'Occhiello.

Dal menu TEXT scegliamo l'opzione TAG, che rende attivo il requester tramite il quale possiamo operare una serie di scelte. Innanzitutto inseriamo il nome, Occhiello, appunto. Nella parte destra del requester notiamo un elenco di possibili opzioni che vanno dalla scelta della fonte-carattere al tipo di spaziatura, dalla giustificazione al colore. Tutte le scelte avvengono tramite box di dialogo. Per l'Occhiello scegliamo la fonte Tyme di corpo 18, ovvero quella che più si avvicina alla fonte dei titoli di *Commodore Gazette*. Altri elementi caratteristici di questo Tag sono il nero e la giustificazione a sinistra. Dopo aver operato tutte le scelte necessarie, immagazziniamo il tutto puntando il mouse sulla casella ADD.

Con questo stesso criterio procediamo per gli altri elementi della pagina, cercando di restare quanto più fedeli possibile all'originale. Si tenga presente che è possibile cambiare in ogni momento le caratteristiche di ciascun Tag, ottenendo la variazione automatica di tutti gli oggetti associati.

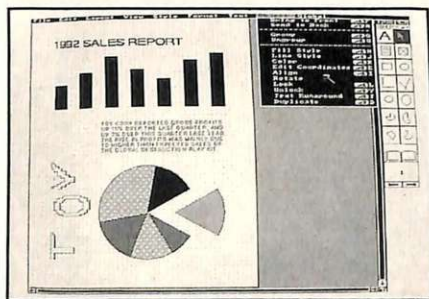
Un'altra caratteristica di una pagina della nostra rivista è la costante presenza del logo di *Commodore Gazette* e del numero di pagina, in basso, nell'angolo esterno della pagina. Questi elementi possono essere convenientemente fissati mediante la creazione della pagina Master, nella quale vengono registrati gli elementi invariabili della pagina. La posizione del logo e del numero di pagina cambia per le pagine pari e quelle dispari, ed è proprio pensando a queste differenze che i programmatori hanno dotato *PageStream* di *due* pagine Master, quella di sinistra e quella di destra. Nel Toolbox c'è un'icona che raffigura un libro aperto: puntando sulla pagina di sinistra attiviamo la pagina Master di sinistra. Il programma conferma, facendo apparire il simbolo LM appena sotto l'icona a forma di libro.

Ora costruiamo il logo della rivista in un punto qualsiasi della pagina (penseremo poi a spostarlo nella posizione definitiva). Prima di tutto attiviamo l'icona Testo, rappresentata da una grande A. Il puntatore si trasforma opportunamente, consentendoci di collocare il cursore in qualsiasi punto della pagina. Scriviamo quindi la parola /COMMODORE e attiviamo l'icona oggetto, rappresentata da una freccia orientata verso l'angolo di sinistra. Puntiamo sulla parola /COMMODORE e premiamo il pulsante di sinistra del mouse. Compaiono otto quadratini che circondano la scritta indicando che l'oggetto è selezionato; d'ora in poi possiamo spostarlo, variarne le dimensioni e operare diverse trasformazioni. Dal menu STYLE selezioniamo la voce FONT/POINTS che ci permette di cambiare lo stile e le dimensioni dei caratteri. Scegliamo la fonte *Helv* a 12 punti che si avvicina abbastanza alla forma del logo originale. Dopo l'OK, la scritta cambia aspetto. In un altro punto della pagina scriviamo la parola GAZETTE utilizzando la fonte *Artistic* a 6 punti e l'attributo ITALICIZE che rende attivo il

corsivo. A questo punto centriamo le due scritte. Un'utile funzione ci aiuta durante questa operazione. Selezioniamo entrambe le scritte puntando il mouse mentre teniamo premuto il tasto Shift, e quindi attiviamo l'opzione ALIGN del menu OBJECT.

Ora vogliamo che le due parole diventino un'entità unica. Le selezioniamo entrambe e scegliamo l'opzione GROUP dal menu OBJECT. Dobbiamo ancora rimpicciolire il logo, rendendolo largo poco più di due centimetri e alto circa mezzo centimetro. Con poche operazioni abbiamo creato un logo che risulta abbastanza simile all'originale.

È arrivato il momento di collo-

care il logo al suo posto nella pagina di sinistra. Per avere una visione globale della pagina possiamo farcela mostrare per intero sullo schermo attivando la voce SHOW FULL PAGE del menu VIEW. Attiviamo l'icona OBJECT e selezioniamo la scritta, quindi la spostiamo nell'angolo in basso a sinistra della pagina, a circa due centimetri dal bordo sinistro e a mezzo centimetro dal bordo inferiore. Questa operazione può essere facilitata dall'uso delle linee guida o della griglia di riferimento, opzioni attivabili tramite menu, che permettono spostamenti rapidi e precisi degli oggetti sulla pagina.

È il momento d'inserire il numero di pagina. Attiviamo l'icona testo e posizioniamo il cursore a un centimetro a sinistra del logo. Dal menu LAYOUT richiamiamo la voce INSERT PAGE NUMBER: compare il simbolo LM. Allineiamo il numero di pagina al logo (tramite l'opzione commentata prima) e il gioco è fatto. Non ci resta che ripetere la stessa procedura per la pagina di destra, avendo l'accortezza d'invertire la posizione del numero di pagina rispetto al logo.

A questo punto possiamo comporre la pagina. Andiamo a pagina 42, che è la pagina dove inizia il nostro articolo, tramite l'icona Numero che si trova in fondo al Toolbox. Selezioniamo l'icona Testo e portiamo il cursore a 3,5 cm sia dal bordo sinistro che dal bordo superiore della pagina e inseriamo il testo dell'Occhiello. Un centimetro e mezzo più in basso inseriamo il Titolo. A 9,5 cm dal bordo superiore collochiamo la Firma.

Creiamo la colonna di testo. Dal Menu LAYOUT attiviamo la voce CREATE COLUMN. Nell'apposito requester inseriamo le misure della colonna. Quando diamo l'OK, la colonna compare nella posizione desiderata. Accanto alla colonna sistemiamo il sottotitolo.

Precedentemente avevamo scritto il testo dell'articolo tramite un word processor e l'avevamo memorizzato su disco come file ASCII. Adesso lo richiamiamo tramite l'opzione IMPORT TEXT del menu FILE, e la nostra pagina è praticamente completa, dobbiamo soltanto associare i Tag ai vari componenti della pagina, e tutto andrà a posto. Le opzioni che abbiamo esaminato sono soltanto una parte di quelle rese disponibili da *PageStream* e proprio per questo, prima di passare alle conclusioni, ci sembra opportuno elencare quelle più significative.

Due momenti molto importanti dell'elaborazione di un documento sono sicuramente la scelta del formato della pagina e la stampa delle parti. Per la prima opzione il programma ci viene incontro con ben 11 formati preimpostati, selezionabili all'atto della creazione del nuovo documento. Sono disponibili per esempio i formati Lettera, Tabloid e alcuni formati standard come A3, A4, B5. Se non dovessero bastare, è sempre possibile definire un formato diverso da quelli proposti, rendendo così virtualmente illimitate le dimensioni del documento. Per quanto riguarda la stampa dei nostri lavori, *PageStream* permette di definire numerosi parametri direttamente da programma. Tra i più importanti ricordiamo: la densità di stampa, il numero di copie e pagine da stampare e la stampa speculare per la realizzazione di lucidi. Un'opzione molto importante è quella relativa alla stampa a colori: è possibile chiedere sia la stampa a quattro colori sia la stampa su un foglio separato di ogni colore utilizzato. Una delle possibilità più interessanti è la scelta della scala di stampa. Grazie a questa opzione è possibile vedere stampate su un unico foglio fino a sedici miniature delle pagine preparate. Questa funzione è molto utile per avere una visione d'insieme del lavoro svolto. Oltre alle opzioni di stampa, *PageStream* mette a disposizione 15 driver che supportano vari tipi di stampanti, le immancabili Epson ma anche alcune Hewlett-Packard, tra cui anche la laser e, naturalmente, lo standard Postscript.

Di sicuro interesse è la voce IMPORT del menu FILE che permette di acquisire testi e grafici preparati esternamente al programma. Possiamo leggere file di testo in formato ASCII o Word Perfect, e file grafici in formato Aegis Draw Plus, Deluxe Paint II e IFFILBM.

Per quel che concerne la gestione delle pagine, delle colonne e dei testi, il programma mette a disposizione una vasta gamma di opzioni. È possibile creare e can-

cellare pagine del documento, nonché cambiare la sequenza dell'impaginazione in qualsiasi momento. Una funzione molto importante in un programma di questo tipo è la creazione delle colonne che qui risulta piuttosto semplice e accurata.

Oltre a queste funzioni, che si possono definire primarie, esistono una serie di altre opzioni non meno importanti. Per esempio, la possibilità di gestire lo scorrimento del testo da una colonna alla successiva e l'inserimento del numero di pagina. Utilissime sono le funzioni relative alle linee guida e alla griglia di riferimento, che rendono le operazioni di posizionamento meno difficili e sicuramente più precise.

La visualizzazione a schermo del nostro lavoro è gestita dal menu VIEW. Grazie alle opzioni presenti in questo menu possiamo stabilire in che misura dev'essere mostrata una pagina scegliendo tra sette diversi formati di zoom, più uno regolabile a piacere. Sempre in questo menu troviamo una nutrita serie di funzioni che permettono di scegliere gli elementi da visualizzare sulla pagina di lavoro. Possiamo scegliere se visualizzare le linee guida, la griglia di riferimento, i contorni delle colonne, i grafici e così via.

Ultima, ma non meno importante, è la possibilità di scegliere l'accuratezza della visualizzazione del testo. Grazie a questa funzione si può ottenere un sensibile aumento della velocità di ricalcolo dello schermo a prezzo di una ridotta accuratezza nella riproduzione del testo a video.

Un ventaglio ragguardevole di possibilità ci viene presentato dal menu STYLE che si occupa delle fonti-carattere. Le fonti messe a disposizione dal programma sono dieci, ma ci risulta che siano disponibili sul mercato americano alcuni dischi contenenti fonti aggiuntive. Molto versatile è la gamma di corpi-carattere che possiamo selezionare. Si va dai piccolissimi 3 punti (circa un millimetro) ai giganteschi 216 punti (circa sette centimetri e mezzo), per un totale di 23 diversi corpi selezionabili. Gli stili di stampa, poi, sono veramente tanti (ben 12), tra cui – per citare solo i tre più inconsueti – ricordiamo il BACKSLANT che inclina all'indietro i caratteri, il MIRROR che permette di scrivere in modo speculare, l'UPSIDE DOWN che permette di scrivere sottosopra.

La formattazione della pagina è una delle operazioni più delicate del desktop publishing e in questa operazione *PageStream* se la cava egregiamente. Sono disponibili diversi tipi di allineamento del testo e di giustificazione che consentono molte combinazioni. È inoltre possibile scegliere manualmente lo spazio tra i caratteri e il tipo d'interlinea. Due opzioni molto importanti permettono di effettuare la divisione in sillabe e la correzione automatica dei testi. Le regole e il dizionario su cui si basano le due funzioni sono pensate per la lingua inglese. Tuttavia è possibile aggiungere e cambiare sia le regole per andare a capo sia il dizionario, adattando quindi le funzioni alla lingua italiana (ma a prezzo di un lavoro lungo e insidioso). Un'altra utile opzione permette di stabilire lo scorrimento del testo intorno agli elementi grafici in base a cinque maschere prefissate.

L'ultimo gruppo di funzioni permette di maneggiare e alterare le caratteristiche degli oggetti, stabilendo il loro colore, il tipo di tratto per le linee e il tipo di disegno per gli oggetti pieni. L'opzione più interessante tra queste è ROTATE che permette non solo la rotazione degli oggetti (sia testi che oggetti grafici), ma anche una serie di distorsioni che permettono di ottenere notevoli effetti visivi.

Nonostante tutte le qualità di questo programma, tuttavia, non ce la sentiamo di esprimere un giudizio pienamente positivo. Abbiamo messo alla prova, come scrivevamo in apertura, sia la versione 1.6 che la 1.8, senza riscontrare molte differenze. Con grande rammarico abbiamo notato invece diverse lacune che proprio non riusciamo a perdonare. Innanzitutto segnaliamo un grave difetto di visualizzazione che lascia la pagina "sporca" ogni volta che si usano alcune opzioni. Per tornare alla normalità bisogna cambiare il tipo di vista, con grande spreco di tempo e di concentrazione.

Un altro difetto è legato alla possibilità di sagomare il testo attorno a un'immagine. Questa funzione influenza anche immagini e testi che non c'entrano nulla, costringendo l'utente ad adottare trucchi più o meno complicati per aggirare il "bug". A causa di questo difetto, il Sottotitolo della nostra pagina di prova appare più basso rispetto alla posizione che occupa nella pagina originale, come si vede chiaramente nelle fotografie che accompagnano l'articolo.

Sempre in tema di problemi, abbiamo riscontrato una gestione piuttosto rudimentale delle comunicazioni con la stampante; per esempio, risulta praticamente impossibile interrompere una stampa in corso, per quanto il manuale affermi il contrario. Altri programmi meno sofisticati, come *C1-Text*, permettono di azzerare il buffer nel caso che si debba interrompere la stampa. Insomma, ci sono una serie di problemi più o meno gravi, che inficiano la completezza e la versatilità del programma e che ci obbligano a emettere un giudizio più che discreto, anziché il prevedibile ottimo.

Possiamo solo augurarci che arrivi presto una nuova versione che corregga almeno una parte di questi errori.

F.P.

STARFLIGHT

Sulla scia di Star Trek, arriva un capolavoro dalla lunga gestazione

Computer: Amiga
Supporto: Disco
Prezzo: L. 38.000
Produzione: Electronic Arts
Distribuzione: C.T.O. (Via Piemonte 7/F, 40069 Zola Predosa - 051/753133)

Che cosa facevate otto anni fa? Se già allora possedevate un computer, probabilmente vi stavate destreggiando con "potentissime" macchine da 3,5K... quasi sicuramente passavate il vostro tempo libero intrattenendovi con semplici programmini dalla giocabilità quantomeno dubbia. I giochi che sono in circolazione oggi, allora esistevano a malapena nelle menti dei futurologi e degli scrittori di fantascienza.

Ebbene, nel bel mezzo di quella situazione c'era qualcuno, negli Stati Uniti, che nonostante tutto stava progettando un videogioco incredibilmente dettagliato, con un numero "impossibile" di opzioni e dalla struttura estremamente complessa. Il programma si ispirava apertamente alla serie televisiva *Star Trek*, di cui riprendeva innumerevoli aspetti. Il gioco, intitolato *Starflight*, sebbene nella sua forma puramente teorica, divenne uno dei primi acquisti della neonata Electronic Arts, che aiutò il game designer a rifinire la meccanica di gioco e cominciò a curarne la realizzazione pratica.

Da allora sono passati 96 mesi, l'industria ha presentato e ritirato dal commercio decine di computer e le tecniche di programmazione hanno raggiunto livelli di sofisticazione impensabili, ma *Starflight* è sopravvissuto a ogni rivoluzione tecnologica ed è arrivato nei negozi in una versione pienamente rispondente al progetto originale.

Starflight è ambientato nell'anno 4600, e il pianeta Terra fa parte della Lega Interstellare. La scoperta di una tecnologia aliena ha permesso la costruzione di astronavi capaci di compiere viaggi intergalattici nella più assoluta tranquillità, e la Lega ha cominciato un programma di metodica esplorazione dello Spazio Esterno.

Il giocatore veste i panni di un comandante della Lega, responsabile di una nave spaziale di ricognizione: rifacendosi alla ben

nota introduzione dei telefilm, il suo scopo viene definito come "andare là, dove nessun uomo è mai giunto prima"... Il primo dovere che ci impongono i regolamenti stellari è quello di raccogliere un equipaggio di sei elementi, che svolgeranno rispettivamente i ruoli di capitano, ufficiale scientifico, navigatore, ingegnere, addetto alle comunicazioni e medico: ogni personaggio possiede differenti capacità e attributi fisici, come in un gioco di ruolo, e può appartenere a una delle numerose razze che fanno parte della Lega, moltiplicando così le combinazioni possibili. Ogni membro dell'equipaggio deve venire addestrato al suo compito, ma con giudizio, perché ogni fase dell'addestramento ha un costo ben preciso, che andrà a intaccare il magro budget messo a disposizione dal comando.

Quello del denaro è un problema molto sentito in *Starflight*, dove l'avarissima Lega Stellare si rifiuta sin dall'inizio di fornire un benché minimo aiuto economico ai suoi stessi equipaggi. Stando così le cose, al giocatore di *Starflight* si apre (letteralmente) un universo di possibilità: in pratica è possibile scegliere di giocare secondo regole del tutto personali, magari dedicandosi alla pirateria o al commercio per raggranellare i crediti necessari a completare la missione affidataci.

L'ambiente di gioco è rappresentato da una carta galattica di dimensioni notevoli, con pianeti inesplorati, tunnel iperspaziali, razze aliene vaganti, piccole guerre locali e mille altri dettagli che rendono la partita affascinante e a suo modo realistica.

È veramente impossibile elencare le moltissime caratteristiche del gioco, i raffinati dettagli e le incredibili situazioni che si incontrano. Il modo migliore per descrivere *Starflight* è forse ricordare *Star Trek* (la serie televisiva), di cui vengono ripresi veramente *tutti* gli elementi. In *Starflight* ci sono gli alieni filosofeggianti, i pianeti sconosciuti da segnalare e registrare per gli archivi della Lega, le battaglie spaziali a suon di faser, missili e scudi direzionali e persino i teletrasportatori, il tutto gestito da un semplicissimo sistema di menu "a pulsante". Grazie alla complessità del programma è facile che una partita media si aggiri sull'ordine delle settimane, ma non saranno settimane noiose.

Certamente non si può dire che *Starflight* sia un gioco originale, tuttavia bisogna ammettere che la cura con cui è stato realizzato fa perdonare questo difetto, rendendo il programma *il* gioco di *Star Trek*, al di là del nome che si trova stampato sulla confezione. Per la stessa ragione, i giocatori che non apprezzano particolarmente la saga fantascientifica più famosa del mondo potrebbero trovare un po' pesante il programma, evidentemente diretto ai fan più sfegatati di Spock e di tutta l'*Enterprise*. **F.R.**

KINDWORDS 2.0

Un word processor semplice ed economico, ma con troppi limiti

Computer: Amiga
Supporto: Disco
Prezzo: L. 109.000
Produzione: The Disc Company
Distribuzione: C.T.O. (Via Piemonte 7/F, 40069 Zola Predosa - 051/753133)

Rieccoci a parlare di word processor. Questa volta siamo alle prese con un pacchetto in grado di fornire prestazioni piuttosto limitate, ma con un prezzo che lo mette alla portata di tutti. Ci limiteremo quindi ad appurare se le poche promesse vengono mantenute, senza perderci in improponibili paragoni con i pacchetti più sofisticati.

KindWords è un editor di testi che si sforza di essere un intermediario molto poco invadente fra l'utente e lo scritto. Anziché mettere a disposizione di chi scrive finestre multiple di testo o elevate possibilità di personalizzazione, i suoi autori hanno preferito ridurre le scelte al minimo e fare della semplicità e dell'immediatezza la loro arma principale. Peccato che, nonostante i diversi miglioramenti inseriti rispetto alla prima versione, il risultato sia ancora un pacchetto dalle capacità molto limitate.

L'editor è dunque completamente WYSIWYG: tutte le caratteristiche grafiche che presenta il documento stampato appaiono già a video. Diventa quindi possibile farsi subito un'idea di come appaiono nel documento le varie fonti-carattere, le intestazioni e le note a piè di pagina, sfruttando così le capacità grafiche dell'Amiga per semplificare e rendere più comodo il lavoro di chi prepara il documento.

Tutte le opzioni offerte dal pacchetto possono essere richiamate attraverso menu a discesa e icone, e non c'è bisogno d'imparare a memoria nessun particolare comando. Non sono comunque presenti molte opzioni per quanto riguarda per esempio la scelta delle fonti-carattere o i comandi di formattazione del testo (impostazione dei margini e rientranze). È possibile fare uso soltanto delle fonti interne dell'Amiga anche se, in compenso, il pacchetto fornisce alcuni "super" driver di stampa che possono

scavalcare quelli di Preferences quando si utilizzano stampanti che non contengono tutti i caratteri richiesti.

Come tutti i word processor, *KindWords* è in grado di leggere e creare file di testo ASCII: sono possibili le classiche operazioni di cut & paste, piuttosto immediate grazie all'uso del mouse come indicatore per i blocchi di testo, e sono anche possibili la ricerca e la sostituzione di parole o gruppi di parole, con eventuale distinzione tra maiuscole e minuscole. Esiste ovviamente una funzione di Undo, molto infelicemente tradotta con "Disfa".

È inoltre consentita la numerazione delle pagine (in numeri arabi e romani, nonché con diversi aspetti tipografici) e l'inserimento di note a inizio e fine pagina, e una particolare opzione ci consente di vedere sullo schermo anche caratteri che normalmente non sono stampabili come i tabulatori, i ritorni-carrello e perfino gli spazi.

Una caratteristica importante per chi desidera produrre stampati contenenti testo e grafica è la possibilità d'importare da altri programmi immagini in formato IFF (per intenderci, quelle di *Deluxe Paint*). L'immagine viene inserita nella posizione indicata dal cursore e, anche se le sue dimensioni non possono essere alterate (contrariamente a quanto afferma il manuale) è possibile ritagliarne una parte.

Esaminiamo infine l'unica caratteristica da autentico "ambiente di lavoro" che ci fornisce *KindWords*: il mail merge. Si tratta della possibilità di scrivere un testo matrice nel quale alcune parti variano in fase di stampa assumendo i valori riportati in un

file di dati a parte. Questo è utile per esempio quando si devono spedire periodicamente lettere, circolari, comunicati e così via, a una serie di nominativi diversi: una volta compilato il documento base (con l'opportuno spazio per i campi variabili), basta collegarlo in fase di stampa con il file dei destinatari, proveniente magari da un database. Manca purtroppo la possibilità di "stampa condizionata" (ovvero che dipenda dal sesso, dall'età o da qualche altra caratteristica del destinatario) di alcuni campi o anche dell'intera lettera (una possibilità offerta per esempio da *geoMerge*).

L'unica novità degna di rilievo che compare nella versione 2.0 del programma è rappresentata dal vocabolario con annesso spell-checker in italiano, piuttosto semplice e non programmabile, ma sicuramente interessante. Non sono molti, infatti, i word processor dotati di questa complessa funzione nella nostra lingua, e, anche se il dizionario in questione è piuttosto limitato e lo spell checker non molto sofisticato, si tratta tuttavia di una piacevole sorpresa. Ci auguriamo che questa funzione diventi una norma, e che anzi vengano aggiunti controlli ortografici più ampi e sofisticati anche in italiano (per chi non lo sapesse, ricordiamo che uno spell checker fornisce un controllo ortografico sulle parole che contiene nel suo dizionario).

Come si può notare, sono presenti tutte le caratteristiche fondamentali di un word processor e forse anche qualcosa di più; inoltre l'impiego è effettivamente molto semplice: il ridotto numero di opzioni diminuisce il rischio di confusione o ambiguità durante il lavoro, e alcuni piccoli accorgimenti rendono semplicissime le operazioni più comuni, evitando inutili fronzoli. È il caso dei comandi di spostamento, che attraverso i tasti funzione, il mouse e le combinazioni Control e Alt permettono di collocare rapidamente il cursore in qualsiasi punto del testo.

Purtroppo, una serie di particolari negativi fa scivolare verso il basso la nostra valutazione di questo word processor, che sin qui sembrava raggiungere la sufficienza. Iniziamo dalla lentezza con cui viene aggiornato lo schermo: già con uno schermo a quattro colori occorre aspettare un bel po' ogni volta che si cambia pagina, ma con una quantità superiore di bitplane le attese diventano tragiche (va comunque detto che una maggiore ricchezza di colori serve soltanto quando si fondono testo e immagini).

Inoltre, l'intero sistema di gestione video risente di gravi difetti di base, a cominciare dai gadget che escono dal margine destro della finestra dell'editor quando questa viene ridimensionata, per finire con lo sfarfallio dei gadget e della stessa finestra a ogni aggiornamento. Forse questi problemi hanno a che fare con alcuni bug relativi alla gestione dei gadget presenti nella versione 1.2 delle funzioni di Intuition, ma comunque resta il fatto che il pacchetto non prova minimamente ad aggirarli e l'utente ne paga le conseguenze soprattutto sotto il profilo dell'accuratezza di realizzazione.

La traduzione in italiano, poi, non è certo di buon livello. Non si era mai sentito dire che un testo o una correzione possono essere "disfati", che un blocco di testo può essere "azzerato" o che tutte le volte che si usa un word processor in realtà si svolge una "seduta" insieme a lui!

Traendo le somme dalla serie di appunti scaturita dall'impiego di *KindWords*, possiamo concludere che questo pacchetto va senz'altro benone per chi non si aspetta grandi cose da un word processor, e non ha nemmeno intenzione di perdere troppo tempo a studiarlo.

A nostro avviso, imparare a utilizzare questo programma in tutte le sue potenzialità richiede un tempo praticamente nullo. Ma se si è disposti a metterci un pizzico d'impegno in più, si può trovare senz'altro di meglio.

D.G. e N.F.R.

DRAGON WARS

Il nuovo gioco di ruolo della Interplay, erede della serie Bard's Tale

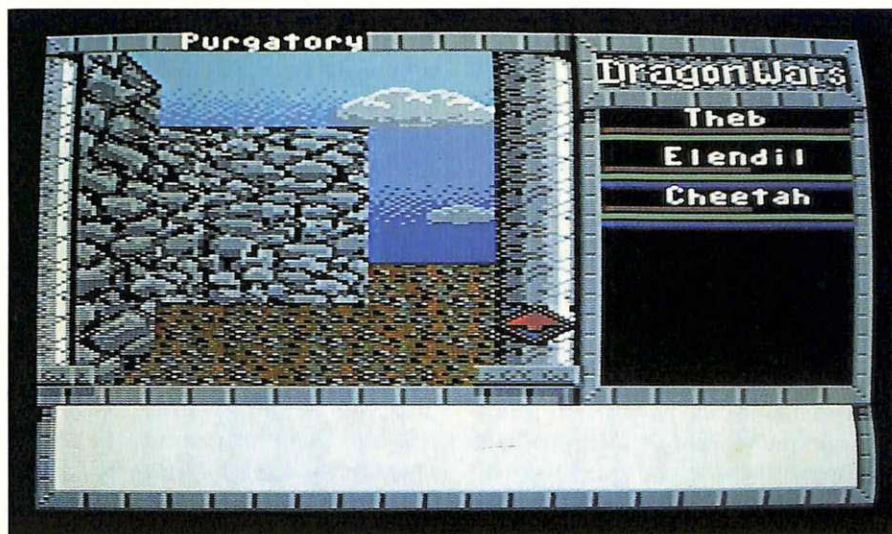
Computer: C-64/128
Supporto: Disco
Prezzo: L. 39.000
Produzione: Interplay
Distribuzione: C.T.O. (Via Piemonte 7/F, 40069 Zola Predosa - 051/753133)

Come passa il tempo... Ricordate *Mindshadow* e *The Tracer Sanction*? Erano le due prime avventure semigrafiche importate ufficialmente dalla MiWa Trading, l'antesignana dei distributori di software indipendenti, e recavano il marchio della Activision. In un angolino dello schermo appariva uno strano marchietto con la scritta "Interplay", ma a quell'epoca nessuno faceva caso agli autori dei giochi.

Oggi Interplay è sinonimo di giochi di ruolo, e con titoli come i vari *Bard's Tale* e *Wasteland* è forse l'unica software house che faccia concorrenza in questo settore alla mastodontica Origin di Richard Garriott. Dopo una piacevolissima variazione con l'acclamato *Battle Chess*, gli autori della Interplay sono tornati alla realizzazione di role playing game, producendo quello che può essere definito un seguito non ufficiale di *Bard's Tale*: un gioco di ruolo dalla trama affascinante che sfrutta schemi ben collaudati e li migliora ulteriormente.

Dragon Wars comincia male: il nostro gruppo di quattro eroi (che possiamo "importare" dai tre *Bard's Tale* o anche, indirettamente, da altri giochi) si è involontariamente macchiato di un crimine contro lo Stato, e si trova segregato su un isolotto noto con il nome di "Purgatorio". Quale sarà mai stato l'orrendo delitto? Forse, è meglio cominciare dal principio...

I nostri eroi stanno cercando il regno fantastico di Dilmun, un paese lastricato d'oro in cui le fontane zampillano ambrosia, e si spingono con la nave in una zona ignota ai cartografi. Giunti nei pressi di un'isola, gli avventurieri prendono terra per riposarsi e trovare qualche informazione, ma commettono l'errore di avvicinarsi alle guardie di re Drake indossando i paramenti dei fruitori di magia. Tutto il gruppo viene quindi punito con l'incarcerazione immediata, come dispone regnante locale, compie un pellegrinaggio ed estrae addirittura la proverbiale spada nella roccia. E alla fine scopre chi si cela dietro lo strano editto di Drake, svelando così un intrigo i cui protagonisti sono nientemeno che gli dei. Naturalmente sappiamo tutti chi dovrà rimettere a posto la situazione...

Se *Bard's Tale* alla fine non diventava altro che una carneficina ambientata in un labirinto e basata solo su apocalittici incantesimi, *Dragon Wars* fa il possibile

un editto di re Drake che ha dichiarato illegale ogni forma di magia.

Come dicevamo, l'inizio di *Dragon Wars* non è certo tra i migliori, ma fortunatamente lo svolgersi degli eventi fornisce interessanti possibilità di avventura: benché i nostri personaggi possano fare ben poco nell'ambito delle Guerre dei Draghi che danno il nome al gioco, dopo l'inevitabile fuga da Purgatorio si aprono parecchi orizzonti, che conducono a un'infinità di microavventure all'interno del gioco vero e proprio. Il gruppetto di avventurieri visita città in rovina, incontra strani culti, capeggia una rivolta per spodestare un

per riportare su una scala più umana questo genere di gioco, recuperando nel frattempo alcuni dei migliori elementi di *Wasteland*. Il risultato è un programma ben bilanciato, semplice da controllare senza arrivare a essere banale, e piacevole da giocare.

Lo schermo principale di gioco riprende il formato di *Bard's Tale*, e offre una finestra grafica nella quale vediamo il paesaggio circostante secondo una prospettiva in prima persona, una finestra con le caratteristiche dei nostri personaggi (rappresentate da barre colorate di lunghezza variabile) e una terza sezione contenente i messaggi di sistema, brevi descrizioni testuali o com-

menti sul gioco. Durante gli incontri, la prima finestra mostra un'immagine animata dei nostri avversari o interlocutori, mentre alla seconda sono riservate funzioni di menu durante i combattimenti. A questa schermata è stata aggiunta un'altra utilissima visuale, richiamabile con un apposito tasto, che consiste in una panoramica dall'alto delle zone visitate, aggiornata costantemente da un dettagliatissimo auto-mapping.

Il sistema di controllo è stato semplificato al massimo, e in effetti è composto da meno di una decina di tasti: quattro servono per muovere il gruppo, mentre gli altri richiamano i menu ramificati; i comandi si impartiscono premendo il tasto relativo alla lettera iniziale dell'opzione desiderata. Questa fa apparire un nuovo menu e così via, sino a quando non si raggiunge un livello di dettaglio soddisfacente. Un esempio può essere il comando "U-sa, (personaggio)1, A-bilità, N-uotare, (direzione)N" (per comodità abbiamo tradotto in italiano le varie opzioni). Il metodo si rivela molto pratico, anche se non sempre rapido quanto si vorrebbe.

Da *Wasteland*, invece, sono state riprese altre due caratteristiche: la prima riguarda l'impiego di un libretto contenente cento e passa paragrafi numerati ai quali il programma fa riferimento per fornire indizi importanti e descrizioni particolarmente lunghe e complesse, mentre il secondo riguarda l'impiego diretto di attributi e abilità per svolgere determinate azioni. Chi conosce *Wasteland* ha senz'altro capito a che cosa ci riferiamo, ma per gli altri è opportuno un esempio chiarificatore. Immaginiamo di trovarci davanti a una porta che ci sbarra la strada: non esistendo un comando particolare per questo tipo di evenienza, possiamo usare l'attributo "Forza" per sfondarla, "Destrezza" per scassinarla o un'ipotetica abilità "Scavare" per passarvi sotto. Un metodo simile si rivela a volte eccezionale e a volte insopportabile, ma bisogna riconoscergli un'indubbia flessibilità.

La meccanica dei combattimenti, importantissima in questo genere di giochi, è sempre quella impiegata per i precedenti titoli della Interplay: il programma tiene conto della distanza relativa dei contendenti e della portata delle armi impiegate, mentre il giocatore seleziona per ogni membro del gruppo un'opzione da un menu "personalizzato". Nella finestra inferiore dello schermo scorrono alcuni messaggi che ci informano sull'andamento della battaglia, finché non viene eliminata una delle due fazioni o i contendenti si allontanano troppo l'uno dall'altro.

La magia, sfruttata in maniera esagerata nei tre *Bard's Tale*, è stata ridotta a "soli" 65 incantesimi, che vengono impiegati costantemente nonostante il loro impatto negativo sulla salute di chi li usa, e nonostante le leggi vigenti nel mondo digitale. Ancora una volta, gli amanti dei duelli "alla mago Merlino" saranno più che soddisfatti.

Elencate tutte le sue caratteristiche, di *Dragon Wars* resta da dire soltanto che, tra i molti giochi di ruolo recensiti in questo numero di *Commodore Gazette*, è senz'altro il più semplice e quello più alla portata dei neofiti. Non tutti amano le travolgenti abbuffate di dettagli di cui sono pieni *Knights of Legend*, *Space Rogue* o *Starflight*, e chi cerca soltanto un po' di azione fantasy, senza grandi problemi tattici o labirintiche complessità, troverà in *Dragon Wars* il titolo per lui.

Un acquisto indispensabile per chi ha già apprezzato i precedenti titoli della Interplay, anche se si rivelerà un po' troppo semplice per i giocatori più smaliziati.

F.R.

KNIGHTS OF LEGEND

Il gioco di ruolo più completo del mondo, ma troppo lento per essere divertente

Computer: C-64/128
Supporto: Disco
Prezzo: L. 21.000
Produzione: Origin
Distribuzione: Leader (Via Mazzini 15, 21020 Casciago – 0332/212255)

Quando si parla di programmi per computer, il tempo assume valori nuovi. In quale altro campo avrebbe senso riferirsi all'anno scorso come a "molto tempo fa"?

Eppure, questo è proprio il caso nel quale ci troviamo parlando di *Knights of Legend*: nel numero di *Commodore Gazette* del settembre '88 avevamo definito "completi" e "realistici" programmi come *Bard's Tale III* o *Wasteland* che, confrontati con questo recente titolo della Origin, appaiono oggi dei primordiali abbozzi di role playing game. Miracoli della tecnologia.

Knights of Legend è forse il miliardesimo gioco di ruolo fantasy prodotto dalla premiatissima software house texana, ma riesce ugualmente a differenziarsi da tutti gli altri grazie a una struttura incredibilmente dettagliata. A detta degli autori, la sua realizzazione ha richiesto un periodo di nove anni, e tenendo in mano il poderoso manuale di 142 pagine che accompagna la confezione non si fa fatica a crederci.

Il gioco è ambientato nel regno fantastico di Ashtalarea, una zona piuttosto vasta, nella quale possiamo trovare montagne, foreste, paludi, fiumi, laghi, colline, pianure e quasi ogni concepibile elemento geografico. Seguendo la moda più recente, il giocatore non ha una missione precisa e ben definita, ma può modificare a piacimento lo svolgersi del gioco

scegliendosi diverse "missioni" fra quelle offerte dai molti abitanti bisognosi di aiuto, sia per sconfiggere un impero di draghi malvagi, sia per arare l'orticello sotto casa. Con una simile struttura di gioco, la longevità del programma è garantita... ma i programmatori non si sono accontentati e hanno ideato un'ulteriore innovazione per catturare l'interesse a lungo termine dei giocatori: *Knights of Legend* si basa infatti su una non meglio identificata "struttura modulare" che, seguendo l'esempio di giochi come *Dungeons & Dragons* et similia, proporrà costantemente nuove possibilità di avventura mediante la pubblicazione di espansioni di vario genere al modulo principale. Nella confezione troviamo infatti un tagliando che, compilato e inviato alla Origin, dà modo di ricevere dischi addizionali contenenti nuove regioni, avventure e personaggi; la stranezza sta nel fatto che questi moduli non saranno messi in vendita nei negozi né pubblicizzati. Una politica aziendale decisamente bizzarra.

Ma non è certo questa l'unica caratteristica sorprendente del programma esaminato: una volta caricato, infatti, le sorprese si susseguono a raffica, lasciando esterrefatto il giocatore abituato alle relative rozzezze di titoli pur notevoli come *Pool of Radiance* o *Ultima IV*.

Già entrando nell'immancabile fase di creazione dei personaggi, c'è da rimanere a bocca aperta: laddove altri prodotti analoghi si limitano a presentare cinque o sei razze e altrettante professioni, *Knights of Legend* stordisce con un menu di ben trentanove possibilità, che arrivano a comprendere addirittura minoranze etniche e "sottospecializzazioni" all'interno di una professione!

Il gruppo di avventurieri (sino a un massimo di sei) si avventura quindi nella città di Brettle, nella quale cominciano tutte le partite, e ancora una volta ci viene il sospetto che il "piccolo" C-64 sia stato nottetempo vittima di una mutazione che l'ha trasformato in un computer del tutto nuovo. In quale altro modo si può spiegare la mappa della città? Fino a oggi avevamo visto schermate a volo d'uccello composte da tasselli grafici, visuali pseudotridimensionali monotone quanto grezze e strane assonometrie ibride, dotate di un certo fascino ma piene di problemi: *Knights of Legend* cancella questo passato per sbalordirci con una visuale aerea-rialzata nella quale ogni casa ha le proprie caratteristiche individuali, i vialetti di ciottoli seguono percorsi non geometrici e le alte torri delle fortificazioni gettano la loro ombra sui fossati e i campi circostanti. L'unico difetto parrebbe l'assenza dei cittadini, ma questo sospetto viene fugato quando ci avviciniamo alle varie costruzioni. Appaiono infatti alcune schermate in soggettiva nelle quali la popolazione locale c'è eccome! Basta vagare un po' per rendersi conto che *ogni personaggio ha la propria fisionomia*, un particolare che non stupisce più di tanto sino a quando non si considera la vastità di Ashtalarea, dove oltretutto abbondano città, villaggi e stanziamenti assortiti.

A mano a mano che il gioco procede, anche il più smaliziato degli utenti non può trattenere qualche esclamazione di stupore. Le occasioni sono molte; quando si impartisce il tradizionale comando "esamina il gruppo", per esempio, anziché una schermata di statistiche numeriche, appare sullo schermo un disegno della nostra banda di avventurieri, in cui sono diligentemente rappresentati aspetto, vestiti e armamenti di ognuno. Come se non bastasse, vengono poi forniti un paio di grafici contenenti dati incredibili come il livello nutrizionale dei singoli, la loro età, la condizione di salute e di stanchezza di sedici diverse aree del corpo.

Dopo molti acquisti e parecchie chiacchiere, viene naturalmente il momento di lasciare la città: ecco cambiare la scala, e i personaggi scelgono se seguire le strade battute o se incamminarsi fra foreste e colline. La scelta ovviamente non è da prendere a cuor leggero, poiché il programma tiene conto di parecchi fattori che influenzeranno l'avventura. Volete sapere quali? In un breve e probabilmente parziale elenco potremmo indicare l'ora del giorno (i brutti incontri si fanno soprattutto di notte, ma attenzione: ogni creatura ha un proprio orario biologico che ne determina i periodi di attività), le condizioni meteorologiche, il periodo dell'anno (in estate ci si può nutrire di frutta), la politica e le leggi della contea da attraversare e, ovviamente, la stanchezza del gruppo.

Stanchezza che, insieme al livello della fame, influisce non poco sulle capacità negli immancabili combattimenti. Queste fasi del gioco si svolgono con un livello di dettagli che ha dell'incredibile, e ogni movimento e azione influisce sul fisico dei nostri eroi. A questo proposito vanno lodati gli autori che, grazie anche alle descrizioni testuali di ogni fase dello scontro, hanno riportato la guerra all'orrore che è in realtà, cancellando i duelli quasi asettici offerti in genere da programmi similari.

Ma torniamo alla meccanica del gioco. In battaglia, ogni personaggio si muove contemporaneamente agli altri ("buoni" e "cattivi"), e la procedura prevede quindi che ogni membro del gruppo venga istruito sull'azione da svolgere prima di ogni turno di gioco. L'inserimento degli ordini avviene tramite una serie di icone abbastanza autoesplicative che permettono di muovere ogni personaggio a tre diverse velocità, di

attaccare e schivare i colpi, di attuare particolari tecniche di difesa e un certo numero di "mosse" nel combattimento a mani nude. I personaggi appartenenti a una certa razza possiedono inoltre il dono del volo, quindi possono decollare, atterrare o spostarsi, sempre a tre velocità diverse. Anche per quanto riguarda la parte "magica" del gioco, i progettisti hanno abbondato in dettagli. Gli incantesimi si basano su una chiave fonetica illustrata nel manuale, e sono "mirati" su razze particolari, a distanze precise e con effetti positivi o negativi.

Arrivati a questo punto, sarete preoccupati per la complessità dei comandi. In realtà, *Knights of Legend* risulta addirittura intuitivo nei controlli, grazie a un'interfaccia grafico-iconica che solo raramente richiede spiegazioni (presenti comunque nel manuale). Purtroppo, un realismo simile non può non reclamare una pesante contropartita, e la mancanza di un qualsiasi commento sonoro non è certo l'unico difetto del programma. In *Knights of Legend* la vera sciagura è rappresentata dai quattro dischi a doppia faccia contenuti nella confezione, sui quali gli autori hanno letteralmente sparpagliato a casaccio i file del programma. Il programma, naturalmente, accede molto di frequente al disk drive, ma l'assenza di ottimizzazione nella disposizione dei file trasforma l'utente in un frenetico disc-jockey il cui divertimento viene quasi annullato dal costante disk swapping cui viene costretto.

Se perdere tre ore per caricare il gioco, riunire il gruppo, uscire dalla città, vincere due battaglie e rientrare alla base non vi crea problemi, acquistate senza indugio *Knights of Legend* e spedite immediatamente la cartolina di richiesta degli scenari. Altrimenti... be', in primavera dovrebbe uscire *Ultima VI*...

F.R.

SOFTWARE HELPLINE

Indiana Jones and the Last Crusade: La soluzione

Nel campus, il professor Jones si prepara a ripartire

Con i vestiti zuppi, il professor Jones era di nuovo a casa. Nonostante quell'abbigliamento poco ortodosso per il clima accademico della Princeton University, Marcus, vecchio e garbato compagno di mille avventure, lo accoglie a braccia aperte e condivide con lui l'euforia per il ritrovamento della Croce di Coronado. Esauriti i convenevoli nell'atrio del college, Indy si rassegna a calarsi nelle vesti del serio e compito professore di archeologia.

Imboccata la porta della palestra per liberarsi della sua "divisa" di eroe scanzonato e provocatore, ha modo di incrociare i guantoni con un pugile che si sta allenando. Appena una breve sessione di allenamento, per ritardare il più possibile l'odiato ritorno alla rigorosa etichetta del mondo universitario; con nuovi impegni che premono, Indy ritorna da Marcus e gli **parla** dell'aspra discussione avuta con un collega, e commenta che il suo rivale **"se l'era voluta"**. Gli ultimi successi meritano un festeggiamento e Indy organizza con Marcus un brindisi assieme ai colleghi; prima però decide di fare una capatina nel suo ufficio. In fondo a un breve corridoio, poco oltre il salone, il professor Jones si trova costretto a **parlare agli studenti** stipati nella saletta d'attesa: li congeda con poche e impacciate frasi di circostanza (le **penultime** che gli passano per la testa).

Chiusasi alle spalle la porta dell'ufficio e immerso nella quiete del suo piccolo mondo familiare, **prende** da un'affollatissima scrivania **le stampe postali, le lettere e i documenti**; sotto la pila di carte trova un **pacco**. Spinto dalla sua proverbiale curiosità, lo **apre** e scopre con somma sorpresa che dentro si trova un vecchio e consunto **diario**; sfogliando le pagine, Indy si accorge che sono riempite dalla sottile grafia di suo padre che ha minuziosamente raccolto sulla carta una marea di appunti su un oggetto affascinante e misterioso... il Santo Graal. Scosso da quel ritrovamento e deciso a non tornare in pasto alla frotta degli studenti in attesa, Indy **apre** la finestra dello studio ed esce nel campus, dove trova ad attenderlo due individui dai modi sospetti.

Senza molti riguardi, i due lo accompagnano in un appartamento lussuosamente arredato, al cospetto del perfido Walter Donovan. Durante un breve colloquio, il professor Jones riceve la notizia della scomparsa del padre. Torna al college e riparte in tutta fretta. **Viaggia** sino alla **casa di Henry**, ovvero il dotto padre di Indiana Jones.

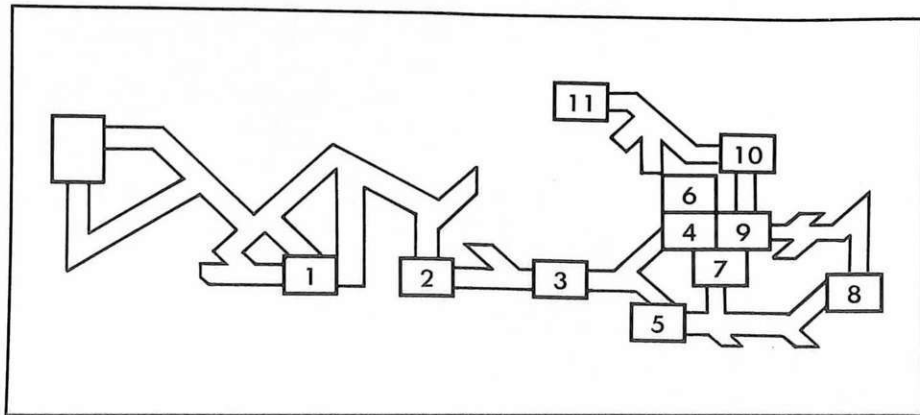
Qualcuno che lo ha preceduto ha messo completamente a soqquadro l'abitazione e Jones decide di proseguire la ricerca. **Tira la libreria** sulla parete di destra, e il crollo delle mensole stipate di antichi volumi gli rivela un pezzo di nastro adesivo. Jones **prende il nastro adesivo**, abbandona l'appartamento e torna al college. Grazie alla finestra aperta, risale nel proprio ufficio: qui, con calma **usa il nastro adesivo su un'ampolla** e trova una **piccola chiave** che Jones è certo possa aprire la serratura di un vecchio scrigno appartenente alla sua famiglia.

Esce ancora una volta dalla finestra e **viaggia** di nuovo verso la **casa di Henry**. Ora sa cosa cercare e conosce l'obiettivo che ha spinto un personaggio misterioso a fare irruzione nell'appartamento del padre. Individuato il prezioso mobiletto nascosto da una tovaglia, **prende la piccola pianta** che serve a tener ferma quest'ultima, quindi **prende la tovaglia** e libera lo scrigno. **Usa la piccola chiave nello scrigno**, poi **prende con sé un vecchio libro** e un **quadro** appeso nella stanza del padre, accanto al letto e a una libreria.

Il professor Jones, dopo aver attraversato la città, ritorna al college. Spiega all'amico Marcus l'urgenza di partire e lo costringe ad accompagnarlo in un **viaggio aereo verso Venezia**.

Suggestioni e peripezie nel fascino della laguna

Ad attenderlo c'è Elsa Schneider, avvenente collega e già assistente di suo padre; in sua compagnia, Jones visita una meravigliosa biblioteca, ricavata fra le mura di una vecchia chiesa. Si guarda in giro, sino a quando una delle numerose stanze che si aprono alla sinistra dell'ingresso, non lo conduce in una sala dove vede un cordone di velluto rosso sorretto da tre paletti. Fa scorrere gli occhi sulle numerose mensole ed **esamina i libri** con estrema cura: lo attrae in particolare un corso di pilotaggio, un testo bizzarro che, tra mille suggerimenti, fornisce anche precise istruzioni su come far decollare un biplano. Jones **prende il manuale di volo** e si annota proprio quelle indicazioni. Come si suol dire *impara l'arte*... Il professore **prende un paletto** (si porta via anche il cordone), estrae il suo piccolo tesoro ed **esamina il diario del Graal**.

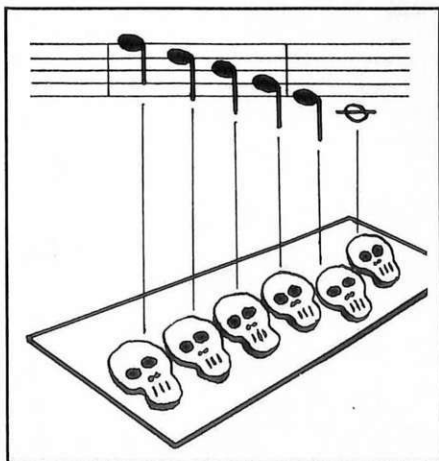
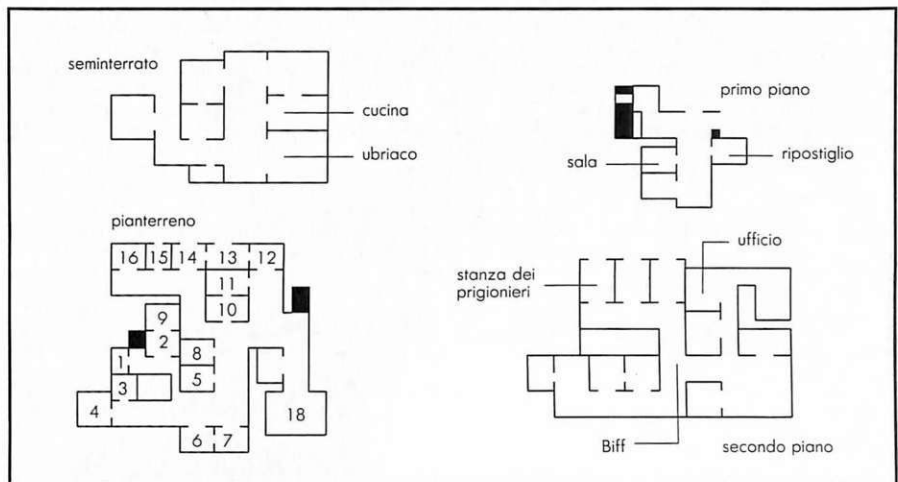
Si addentra poi nel labirinto di sale della biblioteca in cerca di una finestra perfettamente identica a quella riprodotta in una delle pagine del diario; in calce al disegno si legge l'oscuro consiglio del padre. Jones lo segue fiducioso e **legge una delle due iscrizioni** incise sulle colonne ai lati della finestra. Le indicazioni "primo, secondo e terzo" corrispondono ai numeri romani sulle colonne ma anche a quelli effigiati sulle lastre del pavimento. Il professor Jones **usa il paletto metallico sulla**

Le catacombe a Venezia: il nostro punto di partenza è sulla sinistra del disegno

lastra recante il numero rappresentato sul diario, esponendo l'ingresso di un passaggio segreto; non ha neppure il tempo per meditare sulla sorprendente scoperta, che subito la comparsa di un soldato nazista lo costringe a calarsi di corsa nel cunicolo. Di fronte a Indy si parano le buie e antiche catacombe di Venezia. Vinto il primo imbarazzo, il professor Jones non trascura di tracciare una preziosa mappa delle catacombe... e noi la riproduciamo (nella pagina precedente) perché crediamo che Indy non ci avrebbe pensato due volte ad aiutarvi.

Una volta sbucato dal tunnel in basso a sinistra, il nostro avventuriero entra nella caverna numero 1, **prende l'arma** dello scheletro di destra e procede poi sino alla caverna numero 4: quest'ultima comunica direttamente con il sistema fognario della città e Indy, dopo aver **aperto la copertura del tombino**, si ritrova fra i tavolini all'aperto di un ristorante alla moda. Qui, il professor Jones è costretto a infastidire una coppia di innamorati; raggiunge il loro tavolo nell'angolo sulla sinistra, esamina una bottiglia di vino con estrema meticolosità, sotto lo sguardo esterrefatto dei due, e fingendosi un esperto conoscitore di vini, redarguisce il cavaliere per la sua pessima scelta. **Prende la bottiglia**, si allontana tranquillamente dalla coppietta, **usa la bottiglia di vino con la fontana**, riempie quest'ultima di acqua e torna a eclissarsi nel tombino, attirando su di sé i commenti degli astanti.

Giunto nella caverna numero 2, **usa la bottiglia su una torcia** appesa a una parete, per scrostare il fango che la trattiene in una morsa, **prende la torcia** e cade così in una botola che lo porta nei meandri delle segrete.

Indy comincia così la risalita mantenendosi costantemente sulla destra. Lungo il cammino, s'imbatte in un ponte scolpito nella roccia; l'erosione e il persistente sgocciolio dell'acqua, rendono il passaggio sullo strapiombo scivolosissimo e infido. Lì vicino, Indiana Jones scopre una piccola stanza le cui pareti umide sono coperte di misteriose incisioni. **Leggendo le iscrizioni**, Jones trova una conferma alle sue inquietudini e ai suoi sospetti sulla leggenda del Santo Graal: quelle informazioni non devono arrivare alle persone sbagliate. Indy torna indietro, si ferma sul ponte e guarda nel baratro; poi si esibisce come soltanto lui sa fare. **Usa l'uncino sul tappo di legno** dal quale l'acqua continua minacciosamente a sgocciolare sul ponte, quindi **usa la frusta sull'uncino** e libera il tappo facendo cadere sul ponte uno scroscio violentissimo di acqua gelida. Sotto l'improvvisa cascata, il ponte crolla.

Di fronte a Jones, rimane una sola via agibile; si tiene sulla sinistra e raggiunge un'altra caverna. All'interno c'è una consunta scala a pioli che lo porta nella caverna numero 3. Passa in seguito alla numero 5 e da questa, attraverso il tunnel di destra, alla caverna numero 7. Fra le sue pareti,

A ogni teschio corrisponde la nota indicata

La mappa del castello di Brunland, in Austria, dove è prigioniero il professor Jones

campeggia un mastodontico macchinario; Indy **usa il cordone rosso sulla macchina** e poi **usa la ruota** che mette in moto il rugginoso meccanismo. Nella camera si diffonde un odore di stantio e la nota stridula d'ingranaggi corrosi dal tempo e dalla forzata inattività. Il nostro eroe si avvia in direzione della caverna numero 8. I tortuosi passaggi sono ormai asciutti e il labirinto di tunnel risuona di un silenzio sinistro. Davanti ai suoi occhi scorge le sagome dorate di alcune statue misteriose; l'eco dei suoi passi si spegne, Jones si ferma in contemplazione e prende fiato. È tempo di estrarre ancora il suo prezioso tesoro e di **esaminare il diario del Graal**.

Le pagine gli rivelano la posizione in cui disporre gli idoli dorati e Jones **preme le statue** fino a quando non gli si spalanca di fronte un passaggio segreto. L'archeologo inforca l'ingresso buio e sbuca nella caverna numero 9. Per raggiungere la camera numero 11 c'è un solo modo: superare una passerella che il misterioso congegno della numero 7 ha messo in movimento.

Il professor Jones pensa che sia di nuovo il caso di **esaminare il diario del Graal**: i fogli ingialliti gli sono ancora una volta d'aiuto. Lo incuriosisce uno strano riferimento a uno spartito musicale. La sua mente comincia a lavorare a pieno regime. Si fortifica in lui una balzana ipotesi e lentamente assume la consistenza di una teoria plausibilissima: c'è sicuramente un legame tra la posizione delle note sui righi dello spartito e quella dei teschi disposti in una sorta d'inquietante esposizione lungo la parete di fondo della caverna. Nel cervello del professor Jones si accende l'immagine di una tastiera di pianoforte: Indy **preme i teschi** seguendo le note riprodotte sulle pagine del diario (riproduciamo qui a sinistra i teschi che corrispondono alle varie note). Dopo la breve esecuzione si spalanca un altro oscuro passaggio segreto. Jones si ritrova in una nuova sezione delle catacombe...

Ansioso di giungere alla scoperta finale, Jones non ha alcuna difficoltà a districarsi fra le immense caverne della nuova sezione. E finalmente, nella sala finale, ecco davanti a lui la tomba del Crociato. Jones si scuote dallo stupore di quella visione, **apre ed esamina la bara**. E finalmente scopre un'indicazione per localizzare il Graal!

Raggiunge il fondo della camera e trova una grata a sbarrargli la strada; **apre il lucchetto arrugginito**, percorre un maleolente tubo di scarico e riconosce, al termine del cunicolo, un tombino che ha l'aria familiare e rassicurante di una vecchia conoscenza. All'esterno lo attendono i due compagni d'avventura e insieme discutono le mosse successive. La prossima tappa sarà l'Austria.

Nel covo del nemico

Tetre e severe, davanti agli occhi di Elsa e di Jones si ergono le mura del castello di Brunland. Per prudenza, il professore lascia la ragazza a sbirciare il cupo scenario dal finestrino dell'auto parcheggiata poco lontano. Simulando una flemma britannica, Jones bussa al portone principale e viene ricevuto da un impettito maggiordomo in livrea. Indy si presenta come un antiquario inglese. S'intrattiene a **parlare** con il domestico con frasi di rito e dice **le prime frottole** che gli passano per la testa. Presto la conversazione languisce e gli argomenti vengono meno. A Jones viene un prurito alle mani e decide di non indugiare oltre: sferra un diretto al mento del teutonico, imbocca il corridoio di sinistra e si appresta a tracciare una mappa dei locali in cui si trova (in alto).

In una sala, è costretto a ingaggiare un brevissimo dialogo con un uomo delle SS ubriaco fradicio: **"Ciao, come stai?" dice amichevolmente al soldato**. Colta la possibilità di trarre profitto da quella situazione, prosegue: **"Posso offrirti ancora un goccio?"**. Con estrema sorpresa di Jones, la guardia declina l'offerta e allunga all'improvvisato compagno il proprio boccale; un oggetto di cui Indy avrà bisogno tra poco.

Nella cucina del maniero, Jones **usa il boccale con il barile** (pittoresco modo per dire che riempie il bicchiere), e spegne i ciocchi nel caminetto **usando il boccale sui carboni ardenti**. Appena il fumo che avvolge il locale lo consente, **prende l'arrosto** dal fuoco del camino. Ritorna quindi nel salone d'ingresso, sfila di fronte al portone principale e imbocca il corridoio verso il basso. Qui Indy s'imbatte in una guardia e ha la sfrontatezza di spacciarsi per un **uomo della Gestapo venuto per interrogare il prigioniero**. Si presenta addirittura come **il successore di Dietrich**, l'ufficiale che aveva fatto una fine terribile nell'avventura dell'Arca dell'Alleanza. I modi di Jones inducono la sentinella a confidarsi con lui: i due concordano sul fatto che Dietrich **non fosse un brav'uomo**.

Jones ha così la possibilità di spingersi indisturbato nel corridoio e raggiungere un enorme salone: qui trova due armature, e pensando di procurarsi un'arma cerca di **usare l'ascia** di quella sulla destra. L'idea si dimostra pessima e quasi il nostro avventuriero non ci rimette le penne. Abbandona il proposito di rubare un'arma e si limita a **prendere l'uniforme da domestico** da una mensola. Imbocca quindi il corridoio in alto a sinistra e, dovendo fronteggiare un'altra guardia, si trova costretto a **spacciarsi per un venditore di giacche in pelle**. A sostegno della sua buona fede, Jones spiega **che se era riuscito ad arrivare fin lì, era certamente in possesso di un permesso**. La sua parlantina è così convincente che il professore riesce persino a **vendere una giacca per 15 marchi!**

Jones abbandona così il crucco alla sua stupidità, sale le scale fino al primo piano del castello ed

entra nella stanza numero 2. Qui, **apre una cassa e la esamina** e vi scopre all'interno alcune banconote per un valore di 50 marchi. **Usa l'uniforme da domestico**, scende nel corridoio che dà verso il basso e s'imbatte nell'ennesima guardia. Jones è ormai a corto di fesserie da raccontare e non trova altri argomenti che **dare il quadro**. Giunto nella stanza numero 8, Indy può nuovamente tornare a **usare i suoi abiti**.

Apre ed esamina una seconda cassa e trova **un'uniforme nazista**; la **esamina** e scopre una piccola chiave d'ottone. Il più sembra fatto, e Jones decide che è venuto il momento di scendere di nuovo nell'atrio del maniero. Di fronte alla porta di un ripostiglio, **usa la chiave sulla serratura** e, tra le varie divise, **prende l'uniforme grigia**. Corre nella sala numero 8 dove la indossa, **usando l'uniforme grigia**. Attento a non incontrare la sentinella che controlla i corridoi, Jones risale le scale, ma, a mezza strada, si trova davanti una guardia. L'uniforme che indossa gli suggerisce di usare un tono altero e con piglio severo **osserva le pessime condizione della giacca e dei pantaloni del soldato** lo congeda con un inflessibiile "**non farmi perdere tempo!**".

Jones ha quindi modo di entrare nell'ufficio del colonnello Vogel, ma ha la spiacevole sorpresa di trovarsi la strada sbarrata da un poco amichevole pastore tedesco; **dà l'arrosto al cane** e può così accedere all'area riservata. **Prende il trofeo** sullo schedario, **apre un cassetto riservato e prende il pass** che gli rivela la combinazione segreta del forziere situato al piano di sotto. Si precipita sulle scale e le scende a grandi falcate evitando la guardia di pattuglia. Entra in quello che si può ormai definire il suo spogliatoio personale (la stanza numero 8) e qui **usa i suoi abiti** per passare poi alla sala numero 4: questa non è altro che un disordinato deposito di quadri (autentici e falsi), fra i quali il professor Jones riconosce una volgare imitazione della Gioconda. Indy **preme il grande quadro** e la sua mossa gli consente d'individuare il forziere. Jones prende fiato un attimo e, servendosi della combinazione, **apre il forziere**. A bocca aperta, si trova davanti a un quadro che rappresenta il Santo Graal.

Fatto ritorno nell'ingresso, Jones scende le scale che portano alle cucine e **riempie il trofeo con il barile**. Ma, evidentemente, Jones non si trova a suo agio se non torna ogni tanto al suo spogliatoio: raggiunge la stanza numero 8, **usa l'uniforme grigia**, sale ai piani superiori e incontra il mastodontico Biff. Jones rinuncia a qualsiasi approccio aggressivo e decide invece di ricorrere all'astuzia: **dà il trofeo** al biondo gigante che accetta di buon grado e si ubriaca completamente tracannando birra dalla coppa. Jones rimane in paziente attesa e non ha difficoltà ad atterrare l'avversario stordito dall'alcool.

Jones corre quindi alla stanza dei prigionieri, dov'è custodito suo padre, e attiva l'allarme **usando il boccale sui fili elettrici** che escono dalla parte superiore della porta. I due rimangono però bloccati e Jones si trova costretto a **dare il diario del Graal** al colonnello Vogel. Indy e suo padre vengono legati nella sala delle armature, senza che nessuno di loro abbia il tempo di studiare qualche reazione meno arrendevole. Ma è solo un'impressione: la mente di Jones è in realtà un vulcano in continua eruzione. Nella sua testa si accende il lampo di un'idea geniale: **prende Henry** e **tira molte volte le sedie** trascinandole sul pavimento sino al punto nel quale si era abbattuto il precedente colpo di alabarda. Jones **preme l'armatura** e fa cadere la lama dell'alabarda di taglio sulle corde che tengono legati i due ricercatori. I due hanno la strada aperta per la fuga.

Le felici intuizioni di Jones lo portano a localizzare un passaggio segreto che si rivela **premendo la statua** sulla sinistra del caminetto. Sfuggiti alle insidie dei bui labirinti del castello e ritrovatisi nel cortile esterno, Jones e suo padre decidono di raggiungere il confine di Berlino. Jones **usa il sidecar** e, dopo un breve alterco con il genitore, arriva al confine di Berlino. Dopo altre chiacchiere e scazzottate, Indy scopre che il prezioso diario del Graal è finito nelle mani della doppiogiochista assistente tedesca! Spinto da una guardia, Jones si ritrova proprio sulla strada di Hitler; con un improvviso colpo di genio, gli **dà il pass**, che il Führer firma come se fosse uno dei tanti autografi. Ora i nostri amici possono partire per Iskenderun.

In Terrasanta

Giunti all'aeroporto di Berlino, Jones ha modo di fare sfoggio delle sue nozioni di pilotaggio (ricordate il volume recuperato nella biblioteca di Venezia?), **mettendo in moto il biplano** che si trova su una delle piste. Sebbene il decollo non sia dei più facili – anche per l'intervento di due meccanici – il volo si dimostra anche peggiore. Braccato dagli Stukas nemici, il velivolo viene abbattuto. Jones deve rianimare il genitore svenuto e lo trascina fino a una macchina parcheggiata lì vicino; **apre la portiera dell'auto**, sale a bordo e comincia il suo viaggio verso Iskenderun. I due americani passano tranquillamente le varie frontiere e i posti di blocco **dando il pass** firmato da Hitler in persona alle sconcertate sentinelle. Arrivati alle rovine di Alessandretta, Jones si getta a capofitto verso il tempio che custodisce il Santo Graal.

Da questo momento in poi, una forza invincibile (forse il potere del Graal?) non permette più di tornare sui propri passi: l'avventura dev'essere portata a termine, quindi pensateci bene prima di entrare.

Appena dietro al portone, i due archeologi sono accolti dalla macabra visione di una testa che rotola: l'indizio di come il tempio può trattare i suoi visitatori. Nel sontuoso salone principale, il malvagio Donovan (dopo aver crivellato di colpi il corpo di Henry) costringe il professor Jones a esporsi in prima persona alle insidie della ricerca del Graal. Indy affronta l'impresa ritornando con la memoria sia ai consigli del padre, sia ai diagrammi tracciati sulle pagine ingiallite del consunto diario. La sua cerca si alterna così a lucide intuizioni e rocambolesche acrobazie per evitare ogni sorta di trabocchetti: dapprima evita, buttandosi a terra, una mastodontica lama che cerca di spiccargli la testa dal busto, poi è costretto a cimentarsi in un bizzarro "percorso cifrato" (si trova infatti in una sala il cui pavimento è formato da piastrelle siglate con le lettere dell'alfabeto, e deve poggiare i piedi soltanto sopra quelle che formano il nome di Dio, nell'antica lingua dei costruttori del tempio, IEHOVA). Deve poi camminare nel vuoto, sopra un profondissimo abisso, sorretto soltanto dalla propria fede.

Giunto infine nell'ultima camera, viene posto di fronte all'enigma di riconoscere il Graal fra una decina di coppe e calici delle fogge più arcane; per riuscirci deve ricordare le iscrizioni decifrate nelle catacombe di Venezia, nonché il quadro rinvenuto nel forziere del castello di Brunland.

Sceglie la coppa e rimane tremante a guardarla, chiedendosi se è quella giusta. Poi **usa il Graal con l'acqua santa** e ottiene l'approvazione del Cavaliere, posto da secoli a custodia delle coppe.

Torna infine nella sala principale, dove versando sul corpo del padre qualche goccia dell'acqua santa contenuta nel Graal, ridona vita e vigore alle membra ferite del genitore. Padre e figlio sono riuniti sotto il sacro emblema che il mito voleva perduto per l'eternità: sembra l'immagine in dissolvenza di un film a lieto fine.

Ma c'è un teatrale colpo di scena: Elsa Schneider, accecata dall'ira e dall'avidità, afferra il calice e lo porta con sé oltre il Sigillo sacro che delimita lo spazio entro il quale il Santo Graal può esistere. La sua profanazione scatena un apocalittico terremoto che fa precipitare la donna in un baratro.

La terra ha un sussulto che pare interminabile. Jones attende che gli effetti del cataclisma si attenuino, poi **esamina il Sigillo** e vede che il Graal è fortunosamente rimasto in bilico sull'orlo di un costone di roccia. Per recuperarlo, **usa la frusta sul Graal** e lo riprende in un batter d'occhio. Decide però di **darlo al Cavaliere**, suo legittimo proprietario.

Dopo aver trattenuto il fiato per tutto il film, esaliamo finalmente un sospiro di sollievo, vedendo Indiana Jones e suo padre che si abbracciano sullo sfondo di uno splendido tramonto. Sull'avventura del Santo Graal cala il sipario, mentre ci sembra quasi di avvertire l'applauso di una platea immaginaria. Il computer è ancora acceso, e ci scruta con sguardo ammiccante... abbiamo quasi la tentazione di ricominciare l'avventura.

Space Ace: La soluzione

Il gioco è composto di 33 scene accuratamente descritte nel manuale. Faremo riferimento a questa suddivisione per descrivere le mosse. Il momento in cui effettuare una mossa è importante, e verrà segnalato con chiarezza nei casi in cui è veramente critico. Quando non è specificato, si deve muovere non appena il pericolo si presenta. Se ci sono più mosse, si deve aspettare che Ace abbia completato la precedente prima d'immettere la successiva. Consigliamo a tutti di utilizzare la tastierina numerica piuttosto che il joystick, che è meno preciso.

Scena 1: Borf (il cattivo) si avvicina su una piattaforma antigravità. Appena inizia a sparare DESTRA, poi SINISTRA, poi GIÙ.

Scena 2: un robot volante trivella il terreno, subito DESTRA, poi SINISTRA, ancora SINISTRA quando Ace si inginocchia sotto il robot e ancora SINISTRA quando si prepara a saltare.

Scena 3: robot volanti, GIÙ e poi SU.

Scena 4: SU per rallentare l'astronave.

Scena 5: FUOCO non appena compare il serpente.

Scena 6: un enorme braccio meccanico vi sta per schiacciare, subito DESTRA e poi SU per saltare dal ponte.

Scena 7: DESTRA quando la piattaforma ridiscende, e ancora DESTRA non appena Ace atterra sulla piattaforma.

Scena 8: primo mostro viola, GIÙ appena apre la bocca, e poi DESTRA.

Scena 9: secondo mostro viola, stessa cosa, ma GIÙ e poi SINISTRA.

Scena 10: ultimo mostro viola, FUOCO appena vi afferra.

Scena 11: SU appena compaiono i due topi viola.

Scena 12: SU quando compaiono i cani.

Scena 13: DESTRA

Scena 14: SU

Scena 15: sfuggiti ai cani ci si trova in mezzo a due robot pronti a sparare, subito a DESTRA.

Scena 16: SINISTRA.

Scena 17: SINISTRA.

Scena 18: DESTRA.

Scena 19: SU per salire nel centro di controllo di Borf.

Scena 20: da questa scena inizia una lotta con Borf che occupa otto scene senza respiro, bloccate il colpo di Borf premendo FUOCO.

Scena 21: ancora FUOCO per bloccare un colpo.

Scena 22: ancora FUOCO, poi, quando Ace si trova sulla schiena, DESTRA per sfuggire a un altro colpo.

Scena 23: subito FUOCO appena inizia la scena, poi GIÙ per evitare un calcio.

Scena 24: FUOCO per bloccare ancora il colpo di Borf.

Scena 25: Borf fa vibrare la sua asta per due volte, SU e poi GIÙ per evitare i due colpi.

Scena 26: subito DESTRA per evitare l'ultimo colpo di Borf, GIÙ per saltare sulla sua schiena.

Scena 27: SINISTRA quando compare la corda.

Scena 28: Ace salta sulla piattaforma dove è legata Kimberly, nessuna mossa.

Scena 29: quando la piattaforma atterra, DESTRA per sfuggire alla lava.

Scena 30: DESTRA per evitare il raggio.

Scena 31: SINISTRA quando Ace arriva all'altezza dell'altro ponte.

Scena 32: DESTRA.

Scena 33: SINISTRA, poi appena spostato lo specchio DESTRA per evitare il raggio e poi godetevi la scenetta finale! ■

CAD: IL COMPUTER AL SERVIZIO DELLA GRAFICA

Quali sono le differenze tra un programma grafico e un programma di disegno? Ecco una panoramica di tutti i CAD e dei programmi di grafica strutturata disponibili per l'Amiga

di Matthew Leeds

Più di cinquemila anni fa, in Egitto fu inventata la carta, e gli uomini cominciarono subito a servirsene per disegnare. Pennelli, penne d'oca, matite, stilografiche e moltissimi altri strumenti sono stati usati per disegnare, ma negli anni recenti ha cominciato a imporsi uno strumento nuovo, uno strumento che sembra ideale per il disegno di precisione: il computer.

Il mezzo elettronico offre molti vantaggi rispetto ai metodi tradizionali: copie identiche del disegno originale, facile correzione degli errori, automatica catalogazione di tutti i suoi elementi, possibilità d'ingrandire o rimpicciolire a piacere il disegno e molte altre sofisticate funzioni.

Ci stiamo riferendo ovviamente al disegno tecnico e ai relativi programmi per computer, che i produttori chiamano "programmi grafici" per distinguerli dai "programmi di disegno" come *Deluxe Paint*. In un programma di disegno, ogni pixel viene collocato sullo schermo "manualmente", o con l'assistenza di appositi tool di disegno. Si può usare, per esempio, un tool "circle" per disegnare un cerchio sullo schermo, ma tutti i disegni sottostanti andranno perduti senza possibilità di recupero. Inoltre, non si può rimuoverlo senza rischiare di rimuovere anche quello che lo circonda.

Perdipiù, se si desidera ridimensionare il cerchio lo si deve fare ingrandendo la parte di bitmap che lo rappresenta, e il risultato ha spesso un aspetto rozzo e dentellato. Altro particolare da sottolineare: quando si effettua la stampa di un'immagine creata con un programma di disegno, il computer invia alla stampante l'ordine di stampare il disegno come insieme di pixel. Sull'Amiga, una schermata in alta risoluzione (nello standard USA) è composta di 640 x 400 pixel; stampata con una tipica stampante a 9 aghi (con risoluzione di 120 x 120 punti per pollice) il risultato è un'immagine grande circa 13 centimetri per 9. E questo è chiaramente inaccettabile, perché in genere si desidera che l'immagine riempia tutto il foglio. Per aggirare l'ostacolo, la maggior parte dei programmi di disegno ingrandiscono l'immagine dello schermo perché riempia la pagina di stampa. Ma poiché l'immagine originale è una bitmap, questo processo d'ingrandimento dà un risultato "frastagliato", come accade per le immagini sullo schermo.

I programmi grafici seguono una tecnica diversa. Anziché lavorare con uno schermo a risoluzione costante e un'unica bitmap, si servono di una tecnica definita "object-oriented" (basata sugli oggetti) per risolvere i problemi di visualizzazione e di stampa dei programmi di disegno. Ciascun "oggetto" che si colloca sullo schermo, viene immagazzinato in una lista in memoria. Ogni volta che si fa un cambiamento su un oggetto, o che se ne aggiunge uno nuovo, la lista viene aggiornata. Le immagini sullo schermo sono la rappresentazione visiva della lista.

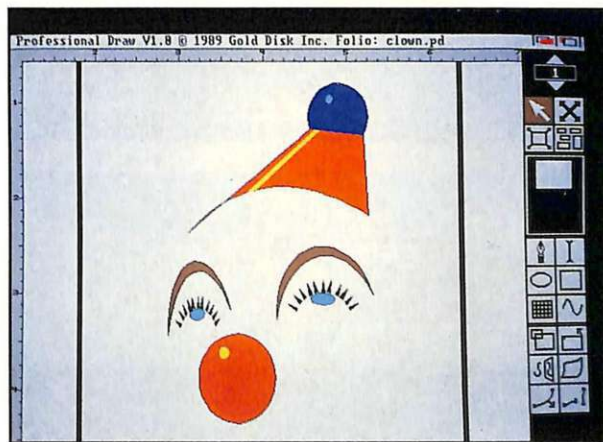
Quando si decide di effettuare la stampa di un'immagine con un programma grafico, l'applicazione controlla il driver di stampa selezionato, per sapere quali sono le risoluzioni disponibili. Poi invia un'immagine bitmap ottimizzata per la stampante. Se si dispone di un plotter, o di una stampante che capisca un linguaggio di "descrizione di pagina" (come una Postscript compatibile, o una Hewlett-Packard che si serva del PCL, Printer Control Language), il programma grafico può inviare una descrizione della pagina anziché una bitmap, e la stampante la

rappresenta utilizzando la massima definizione possibile.

Avere in memoria la lista degli oggetti presenti sullo schermo risolve anche il problema dell'edit. Ricordate il cerchio che avevamo creato con il programma di disegno? Immaginiamo ora di crearlo con un programma grafico. Anziché definire il cerchio punto per punto, dovete inserirne la descrizione indicandone il centro e il raggio. Per modificarlo è sufficiente selezionare il cerchio sullo schermo; il programma ne rintraccia la descrizione nella lista degli oggetti ed evidenzia l'immagine sullo schermo, permettendoci di spostarlo, di variare la lunghezza del raggio, di cancellarlo e così via. Eseguiti i cambiamenti, l'immagine sullo schermo viene aggiornata di conseguenza (il processo di trasferimento dei cambiamenti dalla memoria allo schermo viene chiamato "rendering").

Un serio handicap di tutti i programmi grafici è che l'aggiornamento delle immagini sullo schermo è più lento che nei programmi di disegno. Via via che la lista degli oggetti si allunga, aumenta anche il tempo necessario per ridisegnare lo schermo. Quindi, per lavorare con efficienza (a meno che il disegno non contenga pochissimi elementi), è essenziale che il programma consenta l'uso di una scheda acceleratrice.

Osserviamo esplicitamente, a proposito delle differenze tra progammi di disegno e programmi grafici, che abbiamo tentato di mantenere anche in italiano la differenziazione immediata della lingua inglese, traducendo "Draw" con "grafico" e "Paint" con "disegno". Dovrebbe essere comunque evidente a tutti che un programma che contenga nel nome la parola Draw è in genere un programma grafico (*ProDraw*, *Draw 2000*...) mentre programmi intitolati *DigiPaint* o *Deluxe Paint* sono dedicati al disegno "artistico".

I programmi grafici ricadono fondamentalmente in due categorie: quelli di grafica strutturata e quelli di CAD (Computer aided design, disegno tecnico tramite computer). In alcuni casi è difficile dare una classificazione precisa, ma in generale i programmi di grafica strutturata sono progettati per l'uso artistico, mentre i programmi CAD sono diretti al disegno tecnico e di progettazione. Un programma di grafica strutturata ha in genere la capacità di scambiare bitmap IFF o EPS con programmi di desktop publishing (gli EPS, Encapsulated Postscript Files, sono file che contengono sia la descrizione Postscript di un'immagine sia la sua bitmap, consentendone in ogni caso la rappresentazione a schermo).

Un'immagine semplice e raffinata, creata con ProDraw 1.0

I programmi CAD offrono invece funzioni di dimensionamento e di misura, supporto ai plotter, e molti si comportano anche da database dal momento che mantengono un elenco di tutti gli oggetti utilizzati nel disegno. In genere, i programmi CAD sono più complessi, offrono un numero maggiore di caratteristiche, e costano di più.

I programmi di grafica strutturata

Il primo programma di grafica strutturata per l'Amiga è stato ***Professional Draw 1.0*** della Gold Disk. Offriva una serie di strumenti per la creazione di grafici strutturati che includevano curve di Bezier; linee di vario spessore, colore e tramatura; la possibilità di colorare i contorni o le superfici; output in grado di generare la separazione dei colori per periferiche Postscript compatibili.

Le curve di Bezier vengono usate nei programmi di grafica strutturata e di CAD per tracciare le curve sullo schermo. Ogni curva di Bezier è dotata di almeno quattro punti notevoli: due di ancoraggio, che ne definiscono gli estremi, e due "di controllo" che vengono usati per modificare l'arco della curva. Ogni oggetto che non sia composto di segmenti rettilinei viene creato tramite un insieme di curve di Bezier.

ProDraw 1.0 era afflitto da molte magagne, piccole e grandi. Prima di tutto la velocità di rendering, che con disegni di una certa complessità era bassissima. Inoltre non c'era modo di collegare un testo a una curva, e non si potevano fondere diversi oggetti in uno solo: l'unica possibilità era quella di raggrupparli.

Ho visto all'opera la versione "alfa" di ***ProDraw 2.0***, e molti di questi problemi erano stati superati. Tenete a mente, comunque, che era una versione provvisoria e quindi vi saranno ancora parecchi cambiamenti.

Le maggiori novità sono nel nuovo menu Special. Include le opzioni Align Text with Curve, Text Align Options, Text -> Graphic, Ellipse -> Bezier, Make Compound Object e Blend. La prima opzione (Align Text with Curve) fa esattamente quello che dice il nome: permette d'inserire un testo facendogli seguire la direzione di una linea creata in precedenza. Le numerose opzioni di allineamento del testo (Text Align Options) rendono possibile impaginare il testo a pacchetto, a epigrafe, con bandiera a sinistra e a destra; per il testo sono disponi-

bili inoltre un'opzione di scaling e una di rotazione. Le opzioni Text -> Graphic trasformano il testo in un oggetto grafico, rendendo quindi possibile modificarlo come tale. Ellipse -> Bezier trasforma le ellissi create con il tool Ellipse in un insieme di curve di Bezier modificabili a piacere. Make Compound Object permette di fondere vari elementi in un solo oggetto. Il comando Blend, infine, trasforma un oggetto di Bezier in un altro passando attraverso una serie di "oggetti intermedi" che rappresentano i gradini della trasformazione, il cui numero viene stabilito dall'utente. È un'idea splendida, che permette di creare effetti sorprendenti.

ProDraw 2.0 include inoltre le fonti-carattere create dalla Divisione Computergrafica della AGFA Corporation. Queste fonti-carattere accrescono notevolmente l'utilità del programma e sono le stesse utilizzate da *Professional Page*, un altro pacchetto della serie "Professional" della Gold Disk, dedicato al desktop publishing.

Se lo paragoniamo ai pacchetti grafici disponibili per altre macchine, *ProDraw* non si colloca ai primi posti né come caratteristiche né come funzionalità. Comunque ha arricchito l'Amiga di molte funzioni necessarie, e sta per essere aggiornato. E, soprattutto, è l'unico programma di grafica strutturata nato specificamente per l'Amiga.

La S. Anthony Studios produce tool e utility Postscript per l'Amiga fin dal 1987. I loro programmi della serie *LaserUp!* (*Plot*, *Print*, *Fonts* e *Utilities*) hanno dato per anni ai possessori dell'Amiga un accesso alle periferiche Postscript. ***Plot*** converte file Aegis Draw e Draw Plus in file Postscript da utilizzare con altre applicazioni oppure da inviare a stampanti Postscript compatibili. ***Print*** accetta immagini IFF e produce un output in formato Postscript che permette la stampa a colori o a scala di grigi. Il proprietario della S. Anthony Studios si chiama Scott Anthony, e sta lavorando da due anni a un programma grafico, *LaserUp! Draw*.

LaserUp! Draw possiede alcune

Sopra: LaserUp! Draw, programma di grafica strutturata.
Sotto: un circuito elettronico realizzato con IntroCad

caratteristiche che delizieranno anche i pittori e gli illustratori. Ha un'interfaccia semplice da comprendere, che propone un ben progettato insieme di opzioni e di tool. Un esempio è il comando Place: selezionandolo compare un requester che elenca i tipi di file che si possono "collocare" (place) sulla pagina; file LaserUp! Draw, Generic Postscript, EPS, Aegis Draw, IFF e in formato Raw (non elaborati). Si seleziona il tipo prescelto e dopo il click sull'OK compare un altro requester tramite il quale si può indicare il nome del file. Trovato il file, un altro click sull'OK fa comparire un rettangolo della grandezza del vostro oggetto. Lo si colloca nella posizione voluta, poi si preme il pulsante del mouse: soltanto a quel punto l'oggetto viene disegnato. Ogni oggetto è immagazzinato in memoria, quindi può essere selezionato in ogni momento per essere nuovamente spostato oppure cancellato. Altre caratteristiche importanti sono la conversione di oggetti da un tipo all'altro, il collegamento del testo agli oggetti, la trasformazione di testo in oggetto per poterlo rimodellare, l'autotracing, ovvero la possibilità di trasformare automaticamente oggetti definiti punto per punto (detti "oggetti bitmap") in oggetti strutturati, il comando Blend che trasforma un oggetto in un altro, le opzioni d'ingrandimento e l'opzione Show Genlock (utilizzata generalmente con gli oggetti troppo complicati per l'autotracing: puntando una videocamera sull'oggetto e sovrapponendo sullo schermo l'immagine così ottenuta, l'utente può inserire in un programma grafico un'immagine video). Continuando nell'elenco, troviamo la funzione Export che permette di separare i colori per la stampa offset, d'impostare ingrandimenti o rotazioni, d'includere spot color (cioè di ottenere, in fase di stampa, una parte dell'immagine a un colore e il resto in bianco e nero), d'inviare l'output a dispositivi Postscript, di stampare segni di taglio e crocini di registro, di stampare in positivo o in negativo, di avere l'emulsione di stampa in su o in giù nel caso di stampa con un RIP (Raster Image Processor, o fotounità, un apparecchio simile a una stampante che trasferisce su pelli-

cola immagini o testi descritti da file Postscript).

Il programma ha molte altre capacità, ma preferisco limitarmi a segnalare quelle più importanti o quelle esclusive. Secondo voci recenti, sembra che *LaserUp! Draw* sarà pubblicato dalla Soft-Logik come prodotto parallelo a *PageStream*. Nel momento in cui questo articolo va in stampa il contratto non è ancora stato firmato, ma secondo entrambe le parti in causa c'è già un accordo di massima. Forse *LaserUp! Draw* verrà ribattezzato ***DrawStream***.

C'è un terzo programma di grafica strutturata in arrivo per l'Amiga. Si dovrebbe chiamare ***ProVector***, e ci sta lavorando la Tallesin. In occasione di una fiera, la software house ha fatto vedere una versione alfa del programma ma si è dimostrata riluttante a concedere in visione alla stampa le copie della pre-release. Solo un foglietto illustrativo dava qualche indizio sulle caratteristiche che dovrebbe avere *ProVector*: curve di Bezier, tracciamento delle immagini, 256 colori sullo schermo scelti da una palette di 16 milioni (tramite il processo detto "dithering", che consiste nel collocare vicinissimi l'uno all'altro coppie o quartetti di pixel colorati, dando all'occhio l'impressione di un colore diverso), versioni PAL e NTSC, finestre multiple per eseguire il cut & paste, supporto *AREXX* e Postscript, compatibilità dei file con i programmi di desktop publishing, supporto alle esportazioni IFF comprese le superbitmap. La Tallesin prevede l'uscita di *ProVector* per il maggio '90. Così questo primo scorcio del 1990 vedrà l'uscita di ben tre nuovi programmi grafici per l'Amiga: *Professional Draw 2.0*, *DrawStream*(?) e *ProVector*.

Programmi CAD

Se state cercando un programma poco costoso, tanto per iniziare a fare la conoscenza con il CAD, un buon punto di partenza è ***IntroCAD*** della Progressive Peripherals & Software. Se non volete lavorare a tutti i costi su grafici in dimensioni reali, e non vi occorrono né i layer né la

Sopra: grafici "importati" in IntroCAD da altri programmi. Sotto: uno schema di ingranaggi realizzato con Draw 2000

capacità di importare/esportare file da/a altre applicazioni (al di là della possibilità di salvare su disco come immagini IFF le schermate di *IntroCAD*), allora *IntroCAD* ha tutto quello che vi occorre.

Questo programma mette a disposizione parecchi tool di disegno, tra cui linee, rettangoli, cerchi, archi e disegni a mano libera. In un disegno si possono usare 16 colori e la palette può essere variata come si preferisce. Vengono messi a disposizione righelli e griglie, e si può attivare una funzione "allinea-a" come aiuto nell'allineamento degli oggetti. Disponibili inoltre lo zoom, lo spessore delle linee, le crocette di puntamento e una libreria delle parti. È disponibile l'output per parecchie stampanti e plotter, ma *IntroCAD* non usa le stampanti di Preferences: prima di acquistare il programma, assicuratevi che sia compatibile con la vostra stampante.

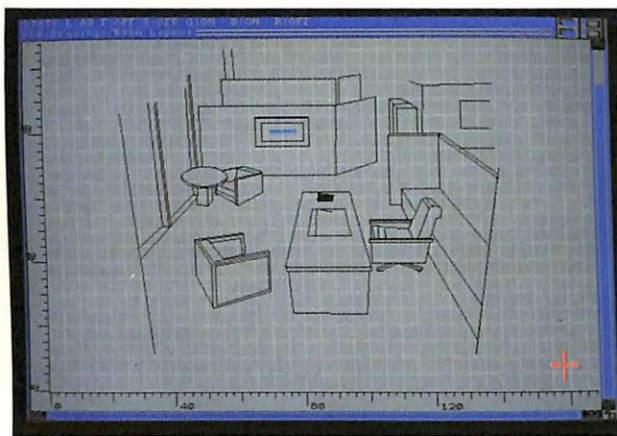
Se le vostre necessità nel settore CAD vanno più in là, potete prendere in considerazione ***Draw 2000***, creato dalla Aegis Development e ora distribuito dalla Oxxi. Le precedenti versioni di questo programma (*Draw* e *Draw Plus*) soffrivano di problemi di velocità e di un'interfaccia non all'altezza dell'Amiga. Ma con *Draw 2000* vi sono stati molti importanti miglioramenti che fanno perdonare quei difetti di gioventù. *Draw 2000* mette a disposizione anche dell'utente Amiga molte importanti caratteristiche tipiche dei CAD, come le finestre multiple, i layer multipli, l'autodimensionamento, la capacità di creare una linea che passi per alcuni punti prefissati, e le cosiddette "parti" (oggetti complessi che vengono visti dal sistema come entità uniche e che possono essere archiviati in memoria o in una libreria su disco).

Il programma opera nelle tre possibili risoluzioni: media, alta e Workbench. In genere si dovrebbe operare in alta risoluzione, ma particolari esigenze di memoria potrebbero rendere necessario lavorare in una risoluzione inferiore. In teoria si può passare da una risoluzione all'altra, ma ho scoperto che *Draw 2000* va in crash se si passa dall'alta risoluzione al Workbench, quando si riporta lo schermo Workbench a piena grandezza. Ho fatto uso di un Workbench in interlace e della

scheda FlickerFixer (per eliminare il flickering) e sospetto che *Draw 2000* non fosse pronto ad affrontare un Workbench in interlace o una finestra di 640 x 400 pixel aperta in quel modo (si veda anche la recensione di *Draw 2000* apparsa sul numero 2/89 di *Commodore Gazette*, n.d.r.).

I layer permettono di lavorare di volta in volta su singole porzioni del disegno. Per esempio, se si sta realizzando il progetto di una casa, si può stendere il disegno delle parti in muratura in un layer, l'impianto elettrico in un altro, l'impianto idraulico in un terzo, e alla fine osservarli tutti simultaneamente. Suddividendo il lavoro in una serie di progetti separati, si vedono più facilmente i risultati dei cambiamenti eseguiti su parti specifiche del disegno. Si può inoltre escludere la visualizzazione di un layer per avere uno schermo un po' più sgombro mentre ci si concentra su qualche aspetto particolare del disegno.

Altre notevoli caratteristiche incluse in *Draw 2000* sono: la funzione di autodimensionamento (la capacità di misurare la lunghezza di un oggetto semplicemente portando il cursore a un'estremità, premendo il pulsante del mouse, e spostandolo all'estremità opposta; il programma dà una rappresentazione della lunghezza misurata, visualizzando una freccia del tipo <– – 3.5 – –>); la funzione di Smoothing, che adatta una curva a un set di punti disegnati sullo schermo dall'utente; la funzione di arrotondamento, che permette d'impostare la precisione del disegno assegnando il numero di decimali da calcolare; i righelli, che possono essere impostati per il sistema decimale, per quello piedi/pollici, o per le unità inglesi (frazioni di pollice).

Ci sono anche diverse opzioni di esportazione. Si può produrre un file da usare con *Modeler 3D* per creare oggetti VideoScape 3D, e inoltre salvare un disegno come immagine IFF per usarlo con *Professional Page*. Si può inoltre stampare su carta una lista di tutte le "parti" utilizzate in un layer o nell'intero disegno. Ovviamente, *Draw 2000* ha anche tutte le caratteristiche standard che ci si aspetta di trovare in un pacchetto CAD: un completo insieme di primitive di disegno; linee di vario spessore, colore e tramatura; strumenti di modifica e correzione; zoom, griglia, righelli, puntatore formato da due linee perpendicolari lunghe quanto i lati dello schermo, e output per plotter e stampanti. Tanto le librerie di parti quanto le impostazioni delle preferenze possono essere salvate su disco e richiamate all'occorrenza.

Draw 2000 (della Aegis) alle prese con il design di interni

Draw 2000 è un buon programma CAD per una fascia di utenza media. Tra i principali limiti segnaliamo la mancanza di combinazioni di tasti equivalenti alle voci dei menu, e d'indicatori di sotto-menu (ovvero quei simboli che seguono una voce indicando subito all'utente se la selezione di quella voce farà comparire a sua volta un sotto-menu). C'è comunque l'opzione "Fast Menu", che fa comparire sullo schermo un toolbox contenente le voci di uso più frequente. L'output verso la stampante non è di alta qualità come quello di altri programmi CAD, ma l'output per il plotter è ottimo. È inclusa anche una versione che dà supporto alla coppia di microprocessori 68020/68881.

A suo merito, diciamo comunque che si tratta di un programma facile da imparare e da usare, e anche piuttosto veloce (a parte la necessità di alcune selezioni di menu piuttosto ripetitive).

Un CAD professionale

Per quanto riguarda la fascia "nobile" dei programmi CAD per l'Amiga, presentiamo un altro pacchetto: *UltraDesign*, della Progressive Peripherals & Software. *UltraDesign* è più sofisticato, più potente, ma richiede anche un maggiore sforzo mentale all'utente. Fornisce un numero illimitato di layer; diciotto fonti-carattere di grandezza definibile dall'utente, la voce Slant (che simula il corsivo), la possibilità di variare il rapporto tra altezza e larghezza dei caratteri a seconda della risoluzione, la creazione di commenti sul disegno (quante sono le parti che lo compongono, quanto costano...), supporto ai file Autocad DXF, Aegis Draw e IntroCAD, supporto ad *AREXX*. Segnaliamo infine che il programma ricorre a un modulo separato e autoconsistente per trasferire disegni separati su una pagina virtuale.

UltraDesign occupa due dischi e per gli utenti che possiedono l'hard disk fornisce anche un programma d'installazione. A prezzo maggiorato, è disponibile anche una versione che supporta l'accoppiata 68020/68881. Una piacevole novità: l'installazione non riscrive la vostra startup sequence, né richiede alcuna assegnazione. Il programma fa ampio uso delle combinazioni di tasti equivalenti ai comandi dei menu (ma manca la solita correlazione mnemonica con i nomi indicati nelle voci dei menu).

Alcuni esempi degli strumenti di lavoro messi a disposizione da *UltraDesign* serviranno per darvi

un'idea delle sue eccellenti capacità. Immaginate per esempio di dover tracciare una perpendicolare a una diagonale. Le operazioni da compiere sono le seguenti: selezionate il punto di partenza della perpendicolare, selezionate poi il tool Perpendicular e infine la linea della quale volete tracciare la perpendicolare, *et voila*: ecco pronta la vostra perpendicolare.

Un altro interessante esempio è il requester Margins, utilizzato per impostare le variabili locali di disegno (che sono 27). Si noti che le 27 variabili riguardano soltanto i margini; ci sono 17 variabili per i parametri relativi alle dimensioni, e altri insiemi simili per ogni elemento del disegno. Se si ha la voglia e il tempo di farlo, si può trasformare il proprio ambiente di lavoro praticamente in qualsiasi modo.

Quasi ogni funzione di *UltraDesign* può essere azionata da comandi dialog. I comandi dialog si usano nei file startup (quelli che vengono mandati in esecuzione alla partenza del programma, un comando dopo l'altro), oppure attraverso procedure *AREXX*.

L'utility PasteUp consente di collocare disegni su una pagina virtuale. I disegni possono essere ridimensionati, ruotati, inviati a un dispositivo di output, e la loro posizione può essere regolata sia tramite tastiera sia tramite mouse.

C'è anche una funzione PosterMaking, che permette la creazione di un foglio virtuale, costituito di un numero qualsiasi di fogli fisici. La stampa può avvenire tramite le stampanti di Preferences; i plotter Roland, HPGL o DMPL; oppure come file PostScript o Encapsulated Postscript.

La qualità degli stampati è eccellente. Personalmente, ho usato il driver Preferences per utilizzare la mia stampante Hewlett-Packard Deskjet, e i risultati sono stati veramente professionali.

L'utility CADverter permette di leggere e scrivere parecchi file CAD dei tipi più frequenti: può caricare file AutoCAD DXF (Data eXchange Format), IntroCAD, Aegis Draw, HPGL (Hewlett-Packard Graphics Language), Roland DXY e UltraDesign. Con file UltraDesign, DXF e immagini IFF può compiere soltanto operazioni di scrittura. I file caricati possono inoltre avere palette propria e impaginazione propria (si ricordino quelle 27 variabili). Queste capacità di conversione sono estremamente utili.

Due immagini realizzate ricorrendo alle numerose "librerie di parti" create dalla Madrigal Residential Designs

I CAD specializzati

A volte un generico pacchetto CAD non mette a disposizione quelle funzioni estremamente specializzate che si desidererebbero. Sono stati quindi creati alcuni CAD "mirati", dedicati alla realizzazione di circuiti stampati, alla creazione di piantine topografiche, di diagrammi di flusso, e così via.

La EAseWare ha realizzato ***HomeBuilders_CAD*** per fornire un aiuto nella progettazione e nella stima dei costi di costruzioni con struttura in legno (oppure in mattoni, o altri materiali da costruzione).

Il pacchetto *HomeBuilders_CAD* contiene due dischi programma (e due copie di backup: è la prima volta che mi capita di vedere copie di backup già pronte) e un manuale veramente ben fatto. Possiede molte delle caratteristiche dei CAD generici, come la rotazione lungo un asse orizzontale (il cosiddetto "pan"), zoom, rotazione, funzioni di edit, autodimensionamento, quotatura. Gli manca una funzione Undo e permette di utilizzare soltanto le stampanti di Preferences.

Fornisce automaticamente proiezioni ortogonali wire-frame anteriori, laterali e posteriori; registra ogni "parte" utilizzata nella costruzione, e stampa un foglio che riassume tutte le singole voci richieste, i costi unitari ed eventualmente un elenco di venditori. Inoltre calcola i costi di scavo e di livellamento. Dai costi generali si possono eventualmente togliere quelli dei materiali, della manodopera e quelli non direttamente attribuibili alla costruzione vera e propria.

Uno dei suoi punti di forza è la libreria delle parti, che elimina la necessità di dover curare ogni dettaglio del disegno. Oltre all'estesa libreria delle parti allegata al programma, se ne può comprare una seconda, la ***HomeBuilders_Library_1***. Se non bastasse, potete procurarvi il programma ***HomeBuilders_Choice***, che per-

mette la creazione di una libreria personalizzata.

Esiste anche un'altra utility, ***HomeBuilders_Print***, utile se occorrono stampati di altissimo livello grafico: permette la stampa in alta risoluzione, guida i plotter HPGL, genera e salva immagini IFF e converte file HomeBuilders_CAD in file Aegis Draw. Lo stesso programma produce inoltre stampati ingranditi fino a 56 centimetri per 86, generando a richiesta la visuale anteriore, laterale o posteriore. Permette inoltre di nascondere sezioni del disegno, di collocare il disegno in un punto qualunque della pagina, e di utilizzare elementi esterni per coprire le eventuali parti non finite.

HomeBuilders_CAD è tortuoso e talvolta difficile da usare, ma fornisce parecchie caratteristiche assolutamente uniche. Se avete la necessità di calcolare costi di costruzione e di generare automaticamente proiezioni ortogonali, e non vi serve un programma di CAD completissimo, allora forse questo è il pacchetto che meglio si adatta alle vostre esigenze.

Utility e librerie

Come per i programmi di disegno, anche per i programmi di grafica strutturata e di CAD esiste un fiorente mercato di dischi contenenti "clip art" e librerie di "parti".

La Madrigal Architectural ha prodotto una serie di dischi contenenti parti, dettagli e schermate da utilizzarsi con programmi di CAD, di disegno e di desktop publishing. Supportano i formati Aegis Draw e IFF. La libreria ***Floor Plan***, per esempio, contiene più di duecento parti: muri, finestre, arredi per bagno, suppellettili per la cucina, caminetti e sezioni di scale. La libreria ***Framing and Foundation Detail*** contiene fondamenta, muri, cornicette, travi, colmi di tetti, e altre parti da costruzione. La libreria ***Interior Design*** include tavoli, divani, sedie, scaffali, librerie, armadi, tolette, cassapanche, mobiletti a muro, scrivanie, armadietti e letti.

La AlohaFonts ha prodotto ***Eclips***, una serie di quattro dischi con 300 esempi di clip art di alta qualità, realizzati nel formato Professional Draw. L'insieme include animali, marchi di fabbrica, suppellettili per ufficio, bordi, cornici, simboli, segni e altri disegnini utili per dare un tocco di fantasia a bollettini, comunicati, volantini, inviti e ogni tipo di materiale stampato.

La Antic Software distribuisce la serie di dischi ***3D Drawings***, uno dei quali è destinato al disegno architettonico. I loro dischi sono in formato Videoscape 3D o Sculpt 3D, ma possono essere convertiti nel formato Professional Draw tramite il convertitore ***InterChange*** distribuito dalla Syndesis.

Per finire, se non siete certi di essere davvero interessati a un serio programma di CAD, ma avete voglia di verificarlo, vi consigliamo il programma ***mCAD***, scritto da Tim Mooney. È disponibile sul Fish Disk 74, un disco che appartiene alla famosa raccolta "Fred Fish" di software di pubblico dominio. Non si tratta di un grande programma di CAD, ma può darvi un'idea di che cosa significa lavorare con un programma di questo tipo. ■

Per ulteriori informazioni contattare direttamente:

AlohaFonts
(Eclips)
POB 2661
Fair Oaks, CA 95628, USA

Anthony Studios
(LaserUp! Draw)
889 DeHaro Street
San Francisco, CA 94107, USA

Antic Software
(3D Drawings)
544 Second Street
San Francisco, CA 94107, USA

C.T.O.
(IntroCAD, L. 109.000)
Via Piemonte 7/F
40069 Zola Predosa (BO)
(Tel. 051/753133)

EAseWare
(HomeBuilders_CAD, HomeBuilders_Library_1, HomeBuilders_Choice, HomeBuilders_Print)
11160F South Lakes Drive,
Suite 305
Reston, VA 22091
USA

Gold Disk
(Professional Draw)
2171 Dunwin Drive, Unit 13
Mississauga, Ontario
Canada L5L 1X2

Madrigal Residential Designs
(Libreria *Floor Plan*,
Libreria *Framing and Foundation Detail*
Libreria *Interior Design)*
POB 2292
Santa Rosa, CA 95405
USA

Oxxi Inc.
(Draw 2000)
1339 E. 28th Street
Long Beach, CA 90806
USA

Progressive Peripherals & Software
(IntroCAD – Distribuzione italiana C.T.O., *UltraDesign)*
464 Kalamath Street
Denver, CO 80204
USA

Syndesis
(InterChange)
N9353 Benson Road
Brooklyn, WI 53521
USA

Tallesin Inc.
(ProVector)
POB 1671
Fort Collins, CO 80522
USA

DAL NOSTRO INVIATO

WORLD OF COMMODORE: ANNO SETTIMO!

Dal World of Commodore Show di Toronto nuovi box di espansione, hard disk e altre periferiche per applicazioni musicali, grafiche e gestionali. Tutti i riflettori sono sull'Amiga, ma il C-64 resiste ancora

di Morton A. Kevelson

Sono di nuovo a Toronto, in Canada. È il primo fine settimana del mese di dicembre, ovvero un periodo dell'anno in cui in genere la gente vola verso sud... io invece sono improvvisamente impazzito e ho spiccato il volo verso nord. Potrà sembrarvi strano, ma insieme a me sono uscite di senno altre 35 mila persone, che hanno deciso di convergere su Toronto: un posto che in media è almeno dieci gradi più freddo della mia città natale, New York.

Il raduno ha lo scopo di celebrare ancora una volta l'anniversario della fondazione della Commodore, la casa produttrice dei nostri computer preferiti. Ma vale davvero la pena di fare questo sacrificio? La Commodore non poteva nascere in aprile, quando il clima è un po' meno rigido?

Be', probabilmente in aprile i suoi fondatori avevano da fare cose migliori che starsene rintanati in un ufficio a fondare società. In ogni caso, è la quarta o forse la quinta volta che mi lascio trascinare in questa follia, e a rischio di essere giudicato un po' matto, ora ve ne parlerò.

Il World of Commodore 1989

Il Canada è un Paese dove la Commodore dichiara di essere il secondo maggior fornitore di microcomputer, e Toronto è la città dove si trova il quartier generale della Commodore canadese. La prima edizione del World of Commodore ebbe luogo nel 1983, l'anno del venticinquesimo anniversario, e fu caratterizzata dalla presenza del VIC-20, da moltissimo materiale di contorno per i vari PET e soprattutto dalla presentazione del C-64.

Quest'anno, invece, benché non mancasse una spruzzatina di C-64 e di C-128, la mostra era dedicata principalmente all'Amiga. E benché la presenza dominante fosse quella dei venditori, c'erano tuttavia abbastanza programmatori e abbastanza nuovi prodotti da garantire l'interesse del pubblico.

A sorpresa, la stella dello spettacolo è stata la terza release di *Demo Reel* della **NewTek**. Finora, *Demo Reel 1* era il programma che mostravo sempre per primo ai nuovi adepti, per dar loro un'idea di quello che sa fare l'Amiga. Non preoccupatevi per la misteriosa assenza di *Demo Reel 2*: semplicemente non è mai stato pubblicato. A quanto si dice, conteneva troppe immagini protette da copyright perché fosse possibile metterlo liberamente in commercio.

Insieme con *Demo Reel 3*, la NewTek ha lanciato sul mercato *DigiView 4.0* l'ultimo aggiornamento del popolare digitalizzatore video. La versione 4.0 di *DigiView* comprende il supporto per l'alta risoluzione (640 pixel) in interlace (400 linee, sistema NTSC) con immagini overscan con 4096 colori presenti simultaneamente sullo schermo. Attenzione: non si tratta del noto modo video HAM (Hold And Modify) che è limitato alle immagini in bassa risoluzione

(320 pixel). La NewTek ha infatti realizzato questo nuovo modo video, per ora unico, ridefinendo la palette (tavolozza dei colori) per ogni linea di scansione.

La NewTek forniva inoltre un disco dimostrativo con molte immagini realizzate nel nuovo modo video. Dopo averle viste, devo ammettere che erano di qualità quasi fotografica (Near Photographic Quality). Al momento in cui scrivo questo articolo, però, *DigiView 4.0* è disponibile soltanto per il sistema americano NTSC. Il programma funziona anche in PAL, ma le immagini non riempiono completamente lo schermo. Comunque la NewTek sta lavorando anche a una versione PAL, che dovrebbe essere disponibile entro breve. Come ideale complemento del programma, veniva presentato il "miglior digitalizzatore video per l'Amiga": DigiView Gold, che permette di digitalizzare e rappresentare a video immagini a 4096 colori e in alta risoluzione. DigiView Gold è specificamente studiato per l'Amiga, e si collega direttamente alla porta parallela. Il prezzo di vendita è di 200 dollari.

Un'esplorazione attenta della mostra ha messo in luce parecchi altri interessanti sviluppi. Aggirandomi nell'enorme area espositiva della Commodore, mi sono imbattuto in Ed Lippert, direttore marketing della **C Ltd.** Lippert era segretamente compiaciuto per essere riuscito finalmente ad assicurarsi un posto nello stand Commodore per *LazerXpress* della C Ltd, a due anni dalla sua uscita sul mercato. La C Ltd esponeva anche un prototipo dei futuri box di espansione Zorro3/1000 e Zorro3/500, destinati rispettivamente all'Amiga 1000 e all'Amiga 500. Questi box forniscono tre slot di espansione "tipo A2000" e altri tre "tipo XT/AT", più un alimentatore indipendente, tutto nello stesso cabinet. In pratica, permettono di far funzionare due schede A2000, una Bridgeboard XT o AT e due schede PC, oppure tre schede A2000 sull'Amiga 500 o 1000. La Zorro3/500 include anche uno slot coprocessore per l'installazione di una scheda acceleratrice 2620.

Il fatto curioso è che Ed Lippert è riuscito a mettere in mostra tutto questo materiale appoggiandosi alla Commodore, e senza avere un proprio stand ufficiale (secondo le ultime notizie questa presenza non deve aver giovato più di tanto all'azienda che sembra infatti aver chiuso i battenti, n.d.r.).

Sopra: Ed Lippert, direttore marketing della C Ltd. Sotto: lo chassis AX-S System creato dalla Spirit Technology

Ancora hardware

La **Spirit Technology** dava dimostrazione del suo chassis AX-S System, un altro tipo di box d'espansione per Amiga 500/1000: un singolo slot A2000, accanto a ben sei slot XT-AT. AX-S System non accetta schede Bridgeboard, né fornisce alcun altro tipo di compatibilità software con l'MS-DOS. In realtà costituisce un'interfaccia che permette a varie schede PC di essere utilizzate tramite un Amiga. Il guaio è che ogni scheda ha bisogno inevitabilmente di uno specifico driver software per essere di qualche utilità.

La Spirit Technology sta pianificando la produzione di parecchi driver di sistema, che consentiranno l'uso con l'Amiga di controller per hard disk, porte seriali, porte parallele e convertitori analogico-digitale. La Spirit prevede di pubblicare come programma di pubblico dominio una "libreria di risorse AX-S" nella speranza che i programmatori indipendenti seguano l'esempio e comincino a creare nuovi driver per i loro sistemi. Avranno successo? AX-S diventerà un punto di riferimento importante nel mercato dell'Amiga? Solo il tempo lo dirà.

Un interessante sviluppo è quello proposto dalla **MichTron**: il prototipo di Fast FAX per l'Amiga. Si tratta di un'unità autosufficiente, ovvero dotata del software necessario per l'interfacciamento con l'Amiga, che mette a disposizione dell'utente la possibilità di ricevere documenti via FAX. Il software prevede anche la possibilità d'inviare nello stesso modo file di testo e IFF. I documenti in arrivo possono essere immagazzinati su disco magnetico, rimandando a un momento successivo la fase di stampa. Manca però la possibilità di effettuare la scansione di un foglio stampato, che non può quindi essere inviato come con un normale FAX. Alcuni sintetici dati tecnici: protocolli di trasmissione V21 (300 bps), V27 (4800/2400 bps), V29 (9600/7200 bps);

compatibilità CCITT gruppo III; interfaccia telefonica RJ11C; interfaccia computer RS232C.

La **Commodore** ha scelto il World of Commodore per il debutto in Canada della A2630, la scheda acceleratrice 68030 a 25 MHz che include il coprocessore matematico a virgola mobile 68882. La A2630, annunciata quest'anno negli Stati Uniti al Comdex, possiede inoltre 2 MB di memoria RAM a 32 bit, espandibile fino a 4 MB.

La Commodore ha presentato inoltre l'A2091, un controller per hard disk SCSI con autoboot, predisposto per alloggiare un'espansione di memoria che può arrivare fino a 2 MB, e l'A2091-40, che al controller aggiunge un hard disk da 40 MB.

Anche la **Great Valley Products** ha puntato molto su una massiccia esposizione di schede hard disk, schede acceleratrici e altre periferiche. Si è data un gran da fare per annunciare la prossima disponibilità della nuova A3001-8MB/80Q: una scheda acceleratrice a 33 MHz basata sul 68030, e dotata anche di coprocessore matematico 68882 a 33 MHz. Oltre alla scheda, nel suo "upgrade kit" la GVP proponeva anche un hard disk Quantum da 40 o 80 MB, con tempo di accesso medio di 11 ms in lettura e 19 ms in scrittura. Il risultato sarebbe migliore di un A2500 con hard disk, secondo i produttori.

La Pre'spect Technics è una società tedesca, che ha recentemente aperto una filiale a Montreal, nel Quebec. Nella sua incarnazione europea, comunque, la società si chiama **Elaborate Bytes**. Al World of Commodore dava dimostrazione della sua linea di controller per hard disk e schede d'interfaccia per hard disk SCSI e ST412/ST506 con l'Amiga. La Pre'spect dichiara che la sua famiglia ALF (Amiga Loads Faster), che comprende prodotti hardware e software, assicura la più alta velocità ottenibile con l'Amiga nel trasferimento dati tra computer e hard disk. Era disponibile inoltre un kit di aggiornamento per l'autoboot per il controller Commodore A2090. Il kit trova posto in uno slot dell'Amiga 2000.

Sopra: anche la Casio crede in un buon futuro per l'abbinamento Amiga/musica. Sotto: Dragon's Lair II

Musica e animazioni

La **Casio**, uno dei leader mondiali nel settore degli strumenti musicali elettronici (oltre che nell'elettronica in generale), si era riservata un ampio spazio espositivo. Un portavoce della società ha affermato che la Casio ha fatto il suo ingresso nel mercato vendendo strumenti musicali, non computer. Di conseguenza, la Casio considera i computer principalmente come congegni adatti a far funzionare le loro tastiere tramite software MIDI. Per provare questo punto di vista, un'intera serie di strumenti musicali veniva "suonata" grazie all'Amiga, senza ricorrere in alcun modo ai suoni campionati interni del computer. È evidente che la Casio giudica l'Amiga un ottimo tramite per riuscire a vendere tastiere e sintetizzatori.

Su versante soft, la **Very Vivid** presenta il nuovo programma *Interactor*, con "hyperFX". Sembra che la parola "iper", di questi tempi, sia diventata onnipresente quando si parla di novità software: secondo il volantino distribuito dalla software house canadese, «*Interactor* (Interactive Graphics Presentation System) con "hyperFX" è un sistema d'animazione che unisce alla potenza dell'hypermedia l'eccitazione del desktop video, e permette di creare database visuali interattivi per realizzare libere associazioni, recuperare informazioni e progettare programmi educativi».

Per chi non lo sapesse, ricordo che la Very Vivid è la creatrice del pacchetto *Mandella*, nonché di alcune dimostrazioni molto interessanti che mescolavano un attore in costume con immagini manipolate tramite l'Amiga, grazie al digitalizzatore *Live!*

Le novità per gli 8 bit

La **Berkeley Softworks** aveva un proprio stand, dove esibiva i vari pacchetti della linea *GEOS*. Aveva anche portato parecchi esemplari delle sue cartucce d'espansione RAM da 512K GEO-RAM per C-64/128. Evidentemente, la cronica carenza dei moduli d'espansione RAM della Commodore (1700/1764/1750) danneggia le possibilità della Berkeley con *GEOS*. Ricordiamo che *GEOS 2.0* e molte delle sue applicazioni software dipendono in grande misura dalla possibilità di attingere a memoria extra, per

funzionare al meglio delle loro possibilità. Il modulo d'espansione della Berkeley non sostituisce completamente i prodotti Commodore: per la precisione, non ha il chip DMA (Direct Memory Access) che rende particolarmente veloce il 1750. Una delle conseguenze è che non funziona con alcuni programmi progettati specificamente per il 1750.

L'unica altra casa produttrice per gli otto bit degna di essere citata è la **Creative Micro Designs**, nota soprattutto per le ROM JiffyDOS, studiate per velocizzare le operazioni di accesso al disco tra C-64/128 e i vari disk drive Commodore, compreso il 1581.

La società pubblicizzava attivamente la sua serie (d'imminente uscita) di hard disk per C-64/128. Queste periferiche, basate sulla tecnologia SCSI, che è ormai l'attuale standard dell'industria, permetteranno prestazioni ad alta velocità sulla porta seriale, purché usate in combinazione con JiffyDOS. Un'interfaccia RAMLink, opzionale, permetterà di collegare l'hard disk anche alla porta parallela per aumentare ulteriormente la velocità di comunicazione. Sono previsti speciali modi operativi per assicurare la compatibilità sia con *GEOS* che col *CP/M*.

Per quanto riguarda i giochi, diciamo subito che la Bethesda Softworks, una divisione della **Media Technology**, annuncia l'imminente distribuzione di *Dragon's Lair II: Escape from Singe's Castle*. L'esatta data di uscita non è ancora stata annunciata, ma sugli schermi della mostra abbiamo potuto ammirare alcune immagini già definitive del programma. A differenza del primo *Dragon's Lair*, i produttori affermano che la seconda puntata della serie «si potrà installare senza difficoltà su qualunque hard disk». Darà inoltre modo di caricare sull'hard disk anche la prima versione, e di unirle l'una all'altra per ottenere un gioco più lungo. Questa volta il giocatore potrà scegliere il numero di vite (da 3 a 5) e il livello di difficoltà. Anche i disegni sono nuovi: molte stanze non si sono mai viste neanche nella versione da bar.

A questa edizione del World of Commodore, la presenza dei computer Commodore a 8 bit era minima. Ma anche se il C-128 è ormai ufficialmente fuori produzione, nello stand Commodore ce n'erano parecchi esemplari in funzione. Perché rinunciare a produrlo, allora? La risposta è che la Commodore è stata incapace di mantenere le posizioni raggiunte dal C-128 sul mercato nordamericano a causa della concorrenza dell'Amiga 500. Con un prezzo di listino di 550 dollari, l'Amiga 500 costa soltanto cento dollari più del C-128D; se si aggiunge anche il fatto che il C-128 è tra i computer Commodore quello che garantisce il minor margine di profitto, non c'è da sorprendersi che la produzione sia stata interrotta.

Tutti gli altri sostenitori degli 8 bit erano gruppi interessati ad aspetti estremamente specializzati, come la Sidfest, un gruppo che si dà da fare per fornire supporti al chip SID del C-64, e che proponeva una speciale offerta composta da alcuni dischi di musica campionata, dalla SID Symphony Stereo Cartridge della **Dr. Evil Laboratories** e dalla coppia libro/programma *Music System for the Commodore 128 & 64, The Enhanced Sidplayer* della Compute! Publications. Per gli ordini postali ci si può rivolgere alla Dr. Evil. ■

Per ulteriori informazioni contattare direttamente:

Berkeley Softworks
distribuita da:
Leader Distribuzione
Via Mazzini, 15
21020 Casciago
(Tel. 0332/212255)

Casio, Inc.
15 Gardner Road
Fairfield, NJ 07006, USA

C Ltd
723 E. Skinner
Wichita, KS 67211, USA
(Tel. 001/316/2676321)
(Fax 001/316/2670111)

Creative Micro Designs Inc.
50 Industrial Drive
PO Box 646
East Longmeadow, MA 01028, USA
(Tel. 001/413/5250023)
(Fax 001/413/5250147)

Dr. Evil Laboratories
PO BOX 3432
Redmond, WA 98073-3432
USA

Elaborate Bytes
BSC büroautomation GmBH
Postfach 400368
D 8000 München 40
Germania Ovest
(Tel. 0049/89/3084152)
(Fax 0049/89/3071714)

Great Valley Products Inc.
225 Plank Avenue
Paoli PA 19301, USA
(Tel. 001/215/8899411)
(Fax 001/215/8899416)

Media Technology Limited
(Bethesda Softworks - Visionary, Ltd)
PO Box 7877
Gaithersburg, MD 20898
USA

MichTron
576 S. Telegraph
Pontiac, MI 48053
USA
(Tel. 001/313/3345700)
(Fax 001/313/3343553)

NewTek Inc.
115 West Crane Street
Topeka, KS 66603
USA

Spirit Technology
220 West 2950 South
Salt Lake City, Utah 84115
USA
(Tel. 001/801/4854233)
(Fax 001/801/4856957)

Very Vivid Inc.
PO Box 127
Stn B. Toronto, Canada M5T-2T3
(Tel. 001/416/6867850)
(Fax 001/416/6868595)

COMPUTERMUSICA: SOUNDSCAPE

UN PRODOTTO CHE HA FATTO LA STORIA DELL'AMIGA

di Massimiliano Lisa

Presentiamo la recensione della versione completa di Soundscape, che comprende Pro MIDI Studio, il campionatore audio Mimetics e i dischi Utilities 1 e 2

La Mimetics è una delle software house che hanno fatto la storia dell'Amiga; il suo programma *Soundscape*, infatti, porta un copyright datato 1986. Il concetto che stava alla base di questo prodotto era interessante: si era pensato di seguire la filosofia della nuova macchina Commodore, e creare un programma che funzionasse in multitasking con qualsiasi altra applicazione e che si servisse delle finestre. Inoltre, non dimenticando che il mondo del software è in continua evoluzione, si era voluto realizzare un programma "aperto", in grado di dar vita a un sistema operativo musicale che si affiancasse a quello dell'Amiga introducendo routine specifiche per la composizione e l'esecuzione di musica e la gestione di suoni campionati. In questo modo, ogni programma applicativo sarebbe stato realizzato con rapidità molto maggiore. Col passare del tempo e con il nascere di nuove tendenze nel campo della computermusica, sarebbe stato sufficiente acquistare periodicamente un disco di aggiornamento, *et voila*: nuove finestre contenenti nuove applicazioni si sarebbero aperte nel proprio ambiente musicale. Ma non solo. *Soundscape* era un sistema operativo aperto a qualunque programmatore: con 150 dollari e una certa conoscenza del linguaggio C si poteva usufruire del pacchetto di programmazione della Mimetics e realizzare applicazioni personali molto più facilmente che non programmando direttamente l'Amiga.

L'ambizione della Mimetics era che *Soundscape* diventasse, per la realizzazione di applicazioni musicali, uno standard seguito anche dalle altre software house, o che perlomeno lo diventasse nel campo del campionamento.

Tutte queste belle aspettative sono state disatte-

se. Se *Soundscape* ha avuto negli scorsi anni un buon successo di pubblico, com'era prevedibile non ha riscosso altrettanto favore tra le software house concorrenti, che hanno pensato bene di confrontarsi con la Mimetics realizzando programmi integralmente nuovi, considerando *Soundscape* non come uno standard da rispettare ma come un punto di partenza da superare. E così, oggi, mentre *Pro MIDI Studio*, il principale programma applicativo di *Soundscape*, e il campionatore sono fermi all'aggiornamento del 1987, altre software house hanno sfornato prodotti decisamente più potenti, sofisticati e facili da usare. A confronto di quanto offre oggi il mercato, i prodotti *Soundscape*, con la loro sterile presentazione a finestre e pochissimi menu, sembrano quasi software di pubblico dominio.

Passiamo ora a un esame ravvicinato del *Complete Soundscape MIDI Music System*, un package importato dalla Soundware (659.000 lire) che comprende diversi prodotti Mimetics in un'unica confezione: *Pro MIDI Studio*, campionatore audio, interfaccia MIDI, *Utilities 1*, *Utilities 2* e *Free Bonus disc*.

Usiamo il sequencer

Ci troviamo dinanzi a un programma suddiviso in numerosi moduli contenuti in altrettante finestre, che ha l'obiettivo di creare sull'Amiga un ambiente musicale completo. La finestra Patch Panel ne è il cuore e contiene i moduli del sistema, che possono essere collegati tra loro a seconda delle esigenze dell'utente. I moduli sulla sinistra sono quelli di input, ossia quelli che forniscono i dati musicali al Patch Panel, e sono: MIDI Mixer, Console Keyboard, MIDI In e Clock In. Quelli sulla destra ricevono i dati musicali in uscita (output) dal Patch panel, e sono: Sampler, MIDI Mixer, Player Piano, Tape Deck, MIDI Out e Clock Out. Attraverso questa finestra si può inoltre caricare o salvare una particolare configurazione di ambiente musicale.

Ma il Patch Panel è molto di più. Tutti i nuovi moduli creati dalla Mimetics (presenti sui dischi *Utilities 1* e *2*) devono essere installati dall'utente in questa finestra che va configurata a seconda delle proprie esigenze. Purtroppo, quello che originariamente doveva essere il punto di forza del programma (l'illimitata espandibilità del sistema modulare) si è rivelato a posteriori un'arma a doppio taglio. Infatti, se da una parte il concetto dell'espandibilità e dell'autoconfigurazione può suonare allettante, dall'altra limita la complessità dei moduli relegandoli in piccole finestre all'interno delle quali le varie funzioni sono costrette all'essenziale. In altre parole, quello che altre software house fanno svolgere a un pacchetto applicativo qui viene svolto da un modulo, e possiamo assicurare che la differenza tra i due termini calza a pennello: dal confronto con i programmi oggi disponibili i moduli di *Soundscape* escono senza dubbio perdenti.

Tape Deck è un sequencer che mette a disposizione un numero illimitato di tracce su cui registrare e di note memorizzabili. L'unico limite è l'ammontare di memoria RAM disponibile. Con 1 MB, se non si utilizzano suoni campionati, si possono registrare circa 14 mila note.

L'inserimento delle note può avvenire attraverso la tastiera dell'Amiga (Console Keyboard) o tramite uno strumento collegato via MIDI. La finestra Console Keyboard visualizza una tastiera limitata a poco meno di tre ottave la cui posizione su un'ideale tastiera da 128 tasti è però ridefinibile dall'utente.

La classica configurazione a finestre del sequencer

Le sequenze possono essere suonate in tempo reale oppure inserite nota per nota. In quest'ultimo caso, le note possono avere una durata variabile tra l'intero brano e 1/384 di nota con una risoluzione di 96 clock per quarto di nota.

Il Tape Deck è paragonabile a un registratore multipista, e in quanto tale dispone di tasti che svolgono funzioni analoghe ad arresto, riproduzione, avanti, indietro, riproduzione dall'inizio del brano, registrazione. È inoltre possibile memorizzare due punti del brano richiamabili istantaneamente (autolocate), e segnaliamo infine le funzioni Punch In e Punch Out, tramite le quali si può riregistrare sopra una parte di traccia delimitata, lasciando inalterato tutto il resto.

Per ogni traccia è possibile impostare dei filtri che permettono di registrare o escludere una serie di eventi MIDI (come program change o pitch wheel). Per esempio si può desiderare che non vengano registrati i movimenti sulla levetta del bender di un sintetizzatore per risparmiare memoria, o perché non si vuole che facciano parte della sequenza che sarà presente sul computer.

Si può registrare una traccia per ogni canale MIDI, che in fase di riproduzione può anche essere riassegnata a un altro canale; l'altezza delle note di ogni traccia può essere aumentata fino a un massimo di 11 semitoni oppure la trasposizione può essere effettuata aumentando o diminuendo fino a quattro ottave; si può ritardare l'inizio dell'esecuzione di una traccia finché non si arriva a un punto particolare del brano, non farla riprodurre o duplicarla. Esistono poi funzioni particolari come Echo, Match, Transpose e Trigger.

Con Echo dev'essere già stata registrata una traccia. Dopodiché si seleziona Play, e il Tape Deck attende l'input da parte dell'utente; a seconda della distanza della nota ricevuta dal Do4, Echo traspone la sequenza dello stesso valore creandone una nuova. Trigger invece, alla pressione di un tasto ricomincia a suonare una sequenza dall'inizio. Match esegue la prima nota della sequenza e attende che sia eseguita anche dal musicista; se la nota è già premuta il Tape Deck continua a suonare, altrimenti s'interrompe fino a quando non la riceve. Transpose effettua la stessa funzione di Echo con la differenza che la modifica della tonalità avviene durante la riproduzione della sequenza da parte del Tape Deck.

Il clock standard di *Soundscape* è di 24 impulsi di clock per quarto di nota, ma si possono usare anche 48 e 96 clock per quarti di nota. Con 24 clock si possono selezionare velocità che vanno da 5 a 600 battute al minuto, con 48 si va da 2,5 a 300 battute al minuto, mentre con 96 si va da 1,25 a 150 battute al minuto. Con 96 clock per quarto di nota l'incremento più piccolo vale 1/384 di nota.

Dopo aver registrato

Ogni volta che si registra musica su una traccia, si crea una sequenza di note e di eventi MIDI all'interno della memoria dell'Amiga. Con la finestra Edit Sequence si può visualizzare la sequenza in formato alfanumerico ed effettuare modifiche. Di ogni nota viene indicato il nome, l'inizio e la durata. Il tempo può essere visualizzato in clock MIDI o in battute e misure, a seconda delle preferenze dell'utente.

Sopra e sotto: la finestra che consente di gestire e modificare campioni e quella per registrare campioni

La possibilità di modificare le informazioni relative alle note con comuni funzioni per copiare, cancellare, inserire, salvare e caricare, esiste anche per tutti gli altri eventi MIDI (Poly Pressure, Control Change, Program Change, Mono Pressure e Pitch Wheel). Se si sono commessi degli errori di tempo nel corso della registrazione, viene in aiuto la funzione Quantize, che permette all'utente di correggere automaticamente il tempo d'inizio di una nota, indicando il numero di clock MIDI desiderati. Questa funzione, però, ha il difetto di non correggere la durata delle note, ma solo la loro posizione.

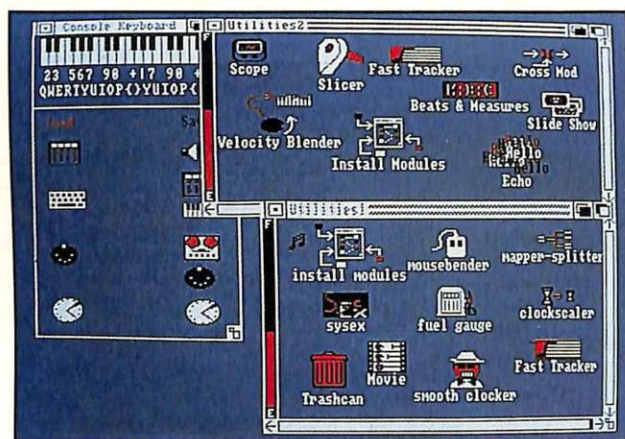
Dopo aver registrato una sequenza e averla modificata attraverso la finestra Edit Sequence, quando si è sicuri di aver raggiunto il risultato desiderato, si può passare alla finestra Edit Song che consente di assemblare tra loro le varie sequenze, creando così il brano definitivo. Va rilevato infine che è presente un programma, *SMUSConverter*, che consente di convertire i file musicali di *Soundscape* nel formato IFF SMUS e viceversa.

Il campionatore

Il *Complete Soundscape MIDI Music System* include anche un campionatore audio, che permette di sfruttare anche la finestra Sampler di *Pro MIDI Studio*. Bisogna subito dire che anche la parte di questo pacchetto relativa al campionamento è piuttosto penalizzata dalla sua origine non certo recente. A suo tempo, infatti, si era scelto di effettuare il collegamento del campionatore attraverso la porta joystick. Oggi invece tutti i campionatori vengono collegati alla porta parallela, e per un ottimo motivo! Il collegamento alla porta joystick limita infatti la massima velocità di campionamento possibile a meno di 15 mila campioni al secondo (e più cam-

pioni al secondo vengono memorizzati, più alta è la qualità). Si consideri che con *AudioMaster II* della Aegis come software e con un campionatore Futuresound si arriva a velocità di 27965 campioni al secondo in stereo e di 44744 in mono. Con una scheda 68020, le prestazioni migliorano ulteriormente arrivando addirittura a 55930 sia in mono sia in stereo.

Vediamo ora il prodotto da vicino. Innanzitutto va detto che la finestra Sample Capture & Edit non regge il confronto con i programmi dedicati oggi disponibili. Le velocità di campionamento possibili sono due, 14080 (14 KHz) e 7040 (7 KHz), e non esiste la modalità stereo. Inoltre *Pro MIDI Studio* non sfrutta appieno le potenzialità del suo campionatore che può arrivare, se viene usato con *AudioMaster II*, a velocità di 8363 e 14914 campioni al secondo, gestendo anche lo stereo. Per quel che riguarda il campionatore, a parte il grave difetto di collegarsi alla porta joystick, va detto che si presenta esteticamente piuttosto bene, con il suo box di plastica grigia e l'adesivo in serigrafia applicato sulla parte superiore. Ha inoltre una caratteristica positiva che parecchi dei più recenti campionatori per la porta parallela non hanno; possiede tre ingressi: microfono, line destro, line sinistro ("line" serve per il collegamento diretto in stereo con registratori, lettori di CD...).

Con il buffer standard da 65K, a 14 KHz è possibile un massimo di cinque secondi di registrazione, mentre a 7 KHz la durata sale a nove secondi. I campioni possono essere trasposti d'ottava, registrati compressi o non-compressi (nel modo compresso vengono esaltati i suoni d'intensità più bassa e attenuati quelli d'intensità più alta) e visualizzati attraverso un oscilloscopio. Con i suoni campionati si possono modificare le forme d'onda, accordare, trasporre, cambiare d'ottava, modificare l'inviluppo, la velocità di risposta alle informazioni MIDI relative a velocità e pitch bender; tutte queste operazioni si possono eseguire tanto sui suoni campionati dall'utente quanto sugli eventuali campioni già presenti su disco (per esempio quelli inclusi in questo stesso pacchetto: organo, piano, archi...).

I moduli di Utilities 1 e 2 visualizzati sullo schermo

La visualizzazione della forma d'onda può essere regolata in modo da mostrare anche soltanto una parte del campione. Si possono ripetere parti di un campione (senza limitazioni) in modo da aumentare la durata del suono. Ogni campione viene riprodotto attraverso le 11 note della Console Keyboard. Per realizzare la gamma completa dei suoni dello strumento, si può trasporre il campione in altre ottave oppure ricampionare la sorgente un'ottava più in alto e una più in basso.

I file possono essere salvati in formato Mimetics oppure IFF.

I moduli aggiuntivi

Come abbiamo già detto, la confezione contiene una serie di prodotti Mimetics, che oltre al disco di *Pro MIDI Studio* comprende i dischi di tre moduli aggiuntivi: *Utilities 1*, *Utilities 2* e *Bonus*.

Utilities 1 contiene i seguenti programmi: *Install Modules*, che aiuta l'utente a configurare il *Workbench* usato da *Soundscape* in modo che i moduli prescelti vengano caricati automaticamente nel Patch Panel; *MouseBender*, che traduce i movimenti del mouse in informazioni MIDI (come pitch bend e aftertouch); *Mapper/Splitter*, che permette di trasformare qualsiasi tipo di dato MIDI in qualunque altro dato MIDI; *Fuel Gauge*, che visualizza graficamente la memoria disponibile; *ClockScaler*, che modifica il rapporto tra il numero d'impulsi di clock MIDI in ingresso e il numero d'impulsi in uscita; *SystemX*, che registra su disco i dati di sistema esclusivo provenienti da apparecchi MIDI; *Frame Counter*, che converte gli eventi delle tracce in un formato secondi/fotogrammi, rendendo possibile capire dove "cade" ogni singolo evento per un'eventuale sincronizzazione con immagini; *SmoothClocker*, che sostituisce il clock originale garantendo una maggior precisione; e infine *Fast Tracker*, che non è un modulo come gli altri ma un programma di pubblico dominio che sostituisce l'usuale sequencer e ha a disposizione un'intera schermata grafica.

Utilities 2 contiene i seguenti programmi: *Fast Tracker* e *Install Modules*, già presenti sul disco 1; *Beats & Measures*, che funziona in combinazione con il Tape Deck, e visualizza il tempo in misure e battute permettendo d'isolare una traccia in fase di riproduzione; *Echo*, che simula l'effetto eco; *Slide Show*, che visualizza immagini realizzate con programmi di disegno suonando nel contempo

brani musicali; *Scope*, che visualizza eventi MIDI nell'ordine in cui li riceve; *Slicer*, che esegue la quantizzazione delle note in tempo reale mettendole a tempo con una specifica battuta; *Velocity Blender*, che permette di modificare i parametri di velocità relativi alle note MIDI mentre vengono suonate; *Cross Mod*, che accetta in input due flussi di note e applica il ritmo dell'uno alla melodia dell'altro.

Il *Free Bonus Disk* contiene i quattro brani vincitori di un concorso tra gli utenti organizzato dalla Mimetics, e una trentina di suoni campionati alcuni dei quali sono veramente di buon livello (per esempio il flauto di Bali).

L'interfaccia MIDI e la documentazione

Gli ultimi elementi che fanno parte della confezione sono l'interfaccia MIDI e la manualistica. L'interfaccia MIDI è ridotta veramente all'essenziale, costituita com'è di uno spinotto seriale RS232 dal quale fuoriescono due cavetti che terminano con due prese identificate dalle etichette In e Out (attaccate con nastro adesivo). Va rilevata l'assenza della presa Thru.

Per quanto riguarda il manuale, oltre al volumetto principale dedicato a *Pro MIDI Studio* (122 pagine) toviamo due Addendum, uno sul campionatore (18 pagine) e uno sulla versione 1.4 del febbraio '87 (12 pagine); e inoltre il manuale del disco *Utilities 1* (30 pagine) integrato da un Addendum (4 pagine), e quello di *Utilities 2* (56 pagine). Come si può notare, la documentazione (in lingua inglese) risulta decisamente frammentata, e non consente certo una lettura organica ed esauriente. Alcune parti dei manuali sono inoltre troppo sintetiche e senza sperimentare il tutto più volte difficilmente si riescono ad afferrare i concetti. Anche perché tutta la manualistica risente di un'eccessiva povertà d'immagini d'esempio.

Il verdetto

Nel 1986 e nella prima metà del 1987 c'era molta eccitazione intorno a *Soundscape*. Unanimemente veniva definito il più completo pacchetto per il campionamento e la gestione MIDI disponibile per l'Amiga. La sua modularità alimentava inoltre l'illusione di un'eterna giovinezza, grazie a continui aggiornamenti con nuovi moduli. Ci si sbagliava.

Il "non-modulo" grafico di pubblico dominio Fast Tracker

La modularità, con le sue finestre ridotte e nessun menu, ha finito per soccombere dinanzi alla qualità delle applicazioni sviluppate in seguito, che hanno sfruttato appieno l'intero schermo dell'Amiga con finestre e requester più evoluti, menu e via dicendo. Ma non è solo questione di presentazione grafica di basso livello e scarsamente user-friendly. Il punto principale è che il livello tecnico complessivo raggiunto oggi è decisamente superiore, per quanto sia il sequencer sia il campionatore contengano tutte le funzioni basilari offerte dai programmi di più recente concezione. Del resto, che il pacchetto-base di *Soundscape* non fosse più attuale lo hanno capito anche alla Mimetics, dal momento che hanno deciso d'includervi un sequencer di pubblico dominio come *Fast Tracker* che con la sua visualizzazione a tutto schermo si rivela molto più facile da usare delle finestre-modulo di *Soundscape* (infatti *Fast Tracker* non viene definito un modulo).

La parola chiave è comunque proprio questa: pubblico dominio. Prodotti di questo tipo hanno senso appunto tra il software di pubblico dominio, nel cui ambito possono svolgere una valida funzione didattica. In caso contrario il prezzo dev'essere davvero basso, come si afferma sulla confezione («...un pacchetto complessivamente di basso costo e facile da usare»). Vendere invece il *Complete Soundscape MIDI Music System* all'esorbitante cifra di 659 mila lire è un'assurdità. Con una cifra quasi analoga si può acquistare un sequencer come *Music-X*, un software di campionamento come *Audiomaster II* della Aegis e un economico campionatore per la porta parallela. E il confronto non è nemmeno proponibile.

Non possiamo inoltre astenerci da un'ulteriore considerazione. Se la soluzione modulare della Mimetics avesse riscontrato un effettivo successo, l'azienda avrebbe sicuramente prosperato maggiormente. Dal 1987, data che contrassegna tutti i copyright dei vari elementi compresi nella confezione, avremmo dovuto vedere parecchi altri moduli aggiuntivi e anche nuove versioni che sostituissero il *Soundscape 1.4*, datato appunto febbraio '87.

Di tutto il package, i prodotti davvero utili sono qualche tool, alcuni suoni campionati registrati su disco e il campionatore, che

può ancora essere interessante per chi ha problemi di memoria (più alta è la velocità di campionamento più memoria si consuma) nonostante sia nettamente superato dai moderni campionatori in grado di collegarsi alla porta parallela. Un prezzo per il quale varrebbe la pena di riflettere dovrebbe quindi aggirarsi intorno alle 250 mila lire. Con qualsiasi cifra superiore potete davvero spendere meglio i vostri soldi dando uno sguardo alla produzione più recente. ■

Per ulteriori informazioni contattare direttamente:

Soundware srl
(The Complete Soundscape MIDI Music System, L. 659.000)
Via Mazzini, 12
21020 Casciago (VA)
(Tel. 0332/222052, fax 0332/228288)

LA POSTA DELLA COMPUTERMUSICA

Non mi è piaciuto...

Ho accolto con vivo compiacimento la nascita della rubrica Computer Musica essendo interessato da un paio d'anni all'informatica e da più lunga data alla musica. Purtroppo il mio primo entusiasmo si è presto trasformato in delusione analizzando il taglio giornalistico della rubrica in questione.

Gli argomenti vengono trattati in maniera superficiale e si dilungano sulla descrizione di software commerciali... Come se ciò non bastasse, si continua a far passare per "musicale" il SID del 64 dimenticando che in base alle specifiche (alta rumorosità, contenuto range dinamico, polifonia max a 3 voci) altro non può che far da supporto ai videogames. Malgrado questi palesi limiti, si pubblicizza a pag. 45 del numero 5/1989 un prodotto definito "decisamente interessante", la SID Symphony Stereo Cartridge di una ditta americana, che possiede la semplice caratteristica di portare a 6 le voci del SID (le 3 residenti più le 3 in cartridge) senza sapere che progetti simili sono già stati pubblicati su riviste di elettronica per poche migliaia di lire (vedi il numero 52 di *Fare Elettronica*)...

La proposta che mi permetto di suggerire prevede un innalzamento qualitativo dei temi trattati, soprattutto dal punto di vista della tecnica di programmazione delle interfacce MIDI collegate ai vari Commodore Computers; pubblicazione dettagliata del protocollo MIDI con le implementazioni più recenti; tecniche di memorizzazione dei dati musicali, della gestione dei sincronismi ed analisi dei MIDI File.

Inoltre molti lettori potrebbero scrivere degli Editor per gli expander poco commercializzati che come si saprà non godono di adeguato software a sostegno. A riguardo il buon Basic, magari il 7.0, rimane perfettamente adeguato non necessitando, quest'ultime applicazioni, di particolari velocità esecutive.

Pino Scambelluri
Reggio Calabria

Non condividiamo il Suo punto di vista. Prima di tutto Le vogliamo far presente che la maggior parte dei fruitori della computermusica non sono in grado di scriversi i programmi da soli, e che quindi sono interessati a recensioni che facciano capire quale software o hardware vale la pena di acquistare. Inoltre, come abbiamo già detto in queste stesse pagine, lo spirito della rubrica è quello di occuparsi delle novità. E dal momento che purtroppo grosse novità in questo campo per C-64/128 non se ne stanno pubblicando, abbiamo già invitato i nostri lettori utenti dei gloriosi 8 bit (e con l'occasione rinnoviamo l'invito) a contribuire con le loro esperienze nel campo inviandoci listati, commenti, progetti hardware... Per quel che riguarda la SID Symphony Cartridge (unica novità per C-64/128 pubblicata negli ultimi tempi) Le ricordiamo che abbiamo specificato "per chi crea musica e la fa eseguire dal SID", e non "per i professionisti del MIDI". Molti utenti fanno eseguire i propri brani esclusivamente dal loro C-64/128, e non tutti sono acquirenti di riviste di elettronica e si cimentano nella realizzazione di kit hardware.

Per quel che riguarda la qualità delle nostre recensioni, non concordiamo con Lei. La invitiamo, ad esempio, a inviarci una recensione su un prodotto come Music-X *più completa della nostra (non ci sembra che l'errore sul clock video sia stato rilevato da nessun altro), oppure ci faccia sapere chi si è accorto che* Deluxe Music *non produce un La440... Il suo invito a parlare di programmazione lo accogliamo invece con favore e lo passiamo a tutti i lettori che desiderano collaborare con noi.*

L'Amiga può fare di più

Prima di tutto vorrei farvi i migliori complimenti per la rivista e per le interessantissime iniziative intraprese come quest'ultima rubrica dedicata alla computer-music, che come campo relativamente nuovo, sta interessando sempre più appassionati tra musicisti professionisti e dilettanti.

Come musicista professionista ed ovviamente felice possessore di un Amiga 2000, ho letto con molto interesse l'articolo di Massimiliano Lisa sul software per la notazione musicale. Mi sembra di dover fare alcune precisazioni; se è vero che il DMCS è rivolto ad una utenza giovane o comunque dalle esigenze limitate, mi sembra altrettanto limitativo il fatto che l'output su carta di un prodotto definito professionale o tale da "offrire una qualità di stampa offset" come il Copyist Professional, sia a mio modesto parere, ancora molto lontano, qualitativamente, da quello offerto da altri programmi altrettanto "professionali" (anche nel prezzo), che girano su altri computer fregiati del titolo di "computer musicali" solo ed esclusivamente per il fatto di essere stati intelligentemente costruiti con interfaccia MIDI interna. Quindi se è vero che bisogna dar atto alla Dr'T di essere stata la prima a convertire il proprio parco software per Amiga, bisogna riconoscere che si è lontani, proprio dal punto di vista della programmazione, da quel tipo di output ad alta definizione per stampanti ad impatto 24 aghi, che offre ad esempio un magnifico DTP: *Page Stream* della SoftLogik.

Direte, cosa c'entra il DTP con la notazione musicale? Ebbene, secondo me, se pensiamo di organizzare le pagine di notazione musicale come "pagine grafiche" (ed in effetti lo sono), sfruttando il principio della "grafica vettorizzata" e non bitmap (come siamo abituati da sempre a vederla), potremmo ottenere risoluzioni su una 24 aghi, driver dedicati permettendo, per male che ci vada 300 X 300 dpi.

Si noti che l'accoppiata *Page Stream*-Nec 24 aghi già offre risultati secondo me sorprendenti (360 X 360 dpi) e superiori a quelli di una stampante laser (300 X 300 dpi), dove la "pulizia" di stampa garantita dal laser non deve lasciarci ingannare e scambiarla come "definizione" di stampa. Anche disponendo di stampanti laser, si avranno ugualmente, sia per le legature che per le linee oblique, quelle evidenti seghettature o zigrinature che non rendono accettabile la stampa.

La verità è che bisognerebbe disporre di unità di fotocomposizione tipo Linotronic, dal prezzo non accessibile, per poter raggiungere ed in qualche caso superare la qualità di stampa che osserviamo sui libri di musica in commercio. Esistono ditte specializzate che effettuano stampe (inviano le "pellicole" per posta) di altissima qualità partendo dal floppy ma, vi invito a fare una ricerca: quante di queste ditte accetteranno dischetti provenienti da un Commodore Amiga e quante offrono già da tempo il servizio per Macintosh?

Comunque, non potendo disporre di un simile servizio, perché non poter fare affidamento sulla nostra 24 aghi, sfruttata, però, al meglio della definizione (grafica vettorizzata)? Ritornando al *Copyist*, perché giudicare eccellenti i risultati ottenuti? Per l'Amiga si può fare molto, ma molto di più.

Stefano Baldoni
Matera

Prendiamo atto delle Sue considerazioni, ma sottolineiamo ancora una volta che l'output su carta prodotto da Copyist *con i driver dedicati è di buon livello. Quanto è apparso sulle pagine della rivista non corrisponde infatti esattamente all'output su carta originale. Il procedimento di riduzione, fotografia, passaggio da pellicola a lastra e da lastra a carta in stampa offset ha infatti prodotto un risultato vicino all'originale, ma non completamente fedele.*

Nome e cognome: ____________________

Indirizzo: ____________________

Città: ____________ Prov: ______ Cap: ______

Desidero ricevere i(l) seguente(i) arretrato(i) di Commodore Gazette:

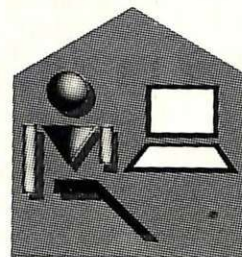
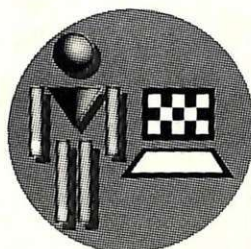
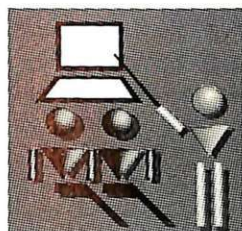
____

Allego la somma di Lire (16.000 ogni numero): ________
intestato alla **IHT Gruppo Editoriale** a mezzo:

☐ vaglia postale (allegando fotocopia della ricevuta)

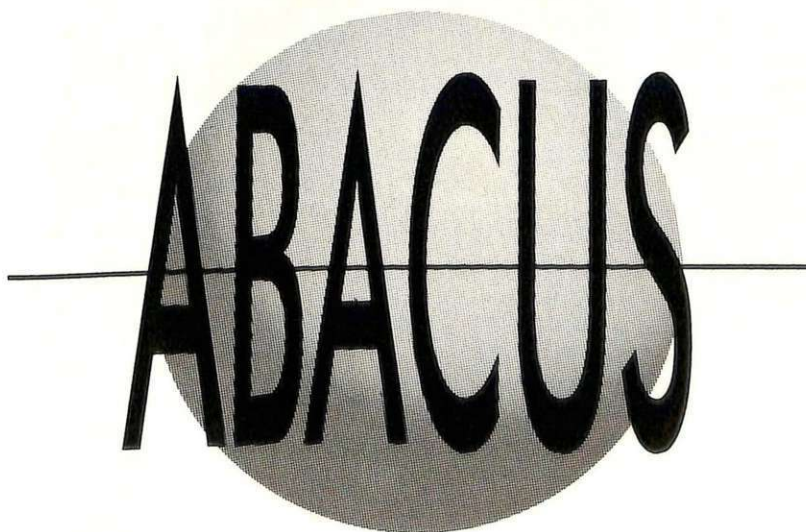
☐ assegno bancario, postale o circolare

Firma: ____________________

Ritagliare e spedire a: **COMMODORE GAZETTE**
Via Monte Napoleone, 9 - 20121 Milano

INDICE DEGLI ARTICOLI SULLA COMPUTERMUSICA

Il seguente indice elenca tutti gli articoli pubblicati sino a oggi in questa rubrica sulla computermusica. È indicato il numero di *Commodore Gazette* sul quale sono apparsi gli articoli. Gli interessati possono richiedere il numero arretrato compilando l'apposita cedola.

Recensioni

Presentazioni

DAL NOSTRO INVIATO

PARIGI! LA TERZA AMIGA DEVELOPERS CONFERENCE EUROPEA

Segreto assoluto sulle novità Commodore dell'anno, che si preannunciano esplosive. Nel frattempo sono in arrivo un simulatore automobilistico e l'emulatore Atari

di Michael Moosleitner

Le eccezionali tempeste che si sono abbattute sull'Europa del nord hanno concesso a Parigi una breve tregua in occasione della terza conferenza europea dei programmatori e dei progettisti di hardware per l'Amiga. Dal 7 al 9 febbraio il fastoso Holiday Inn ha ospitato più di cinquecento tra rappresentanti Commodore, programmatori, progettisti e dirigenti. Sia il numero dei partecipanti (quasi il doppio rispetto al convegno tenutosi l'anno scorso) sia le esigenze e gli orari di chi è abituato a trattare con le macchine più che con le persone, hanno fatto rimpiangere agli altri ospiti dell'albergo le tempeste atmosferiche.

Alcuni top manager Commodore avevano segnato nella loro agendina le date in cui, in base ad anniversari o segnalazioni, erano prevedibili attentati terroristici. Ma pur evitando accuratamente le linee aeree nordamericane, le più brillanti menti che hanno lavorato intorno all'Amiga sono riuscite a ritrovarsi insieme a Parigi. Jay G. Miner, "Padre dell'Amiga", come ama definirsi, è stato il graditissimo ospite d'onore alla conferenza.

Come ogni developers conference, anche questa è stata un evento importante, una rara occasione per tenersi aggiornati sugli ultimi sviluppi del mondo Amiga, studiare le tendenze del settore e incontrare personaggi interessanti.

I discorsi introduttivi, tenuti da Henry Rubin, Kristian Andersen e altri dirigenti Commodore, sono stati estremamente concisi, interessanti e a volte spiritosi. L'Amiga, si è detto, è un sistema maturo che ha raggiunto un notevole potenziale e offre grandi opportunità. Ormai gli Amiga diffusi nel mondo si contano a milioni, ma non è certo questo il traguardo definitivo. La filosofia della Commodore si potrebbe riassumere nella frase: "I bambini di oggi sono gli uomini di domani", e c'è la segreta speranza che una particolare attenzione verso gli utenti più giovani permetterà d'inserire i sistemi Amiga in posizioni chiave entro mezzo secolo. Il prossimo millennio vedrà dunque il sorpasso dell'innominabile Big Blue da parte della Commodore con il suo Amiga 9000?

Le novità "segrete"

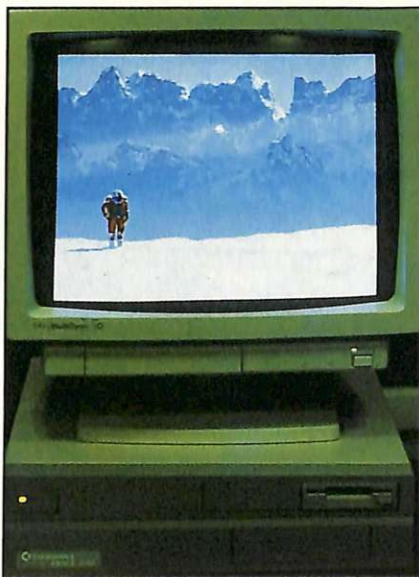
Tra gli annunci "seri", il primo sistema Unix V.4 basato su microprocessore Motorola. È certo che in un futuro non lontano nell'Amiga pulserà un 68040, magari coadiuvato da una scheda Janus 386. A Parigi abbiamo avuto la possibilità di toccare con mano e vedere alla prova materiale appena uscito dai laboratori Commodore della Silicon Valley. I prototipi erano guardati a vista, giorno e notte, chiusi in una stanza con codice d'accesso elettronico che veniva cambiato continuamente. Come tutti coloro che hanno avuto la possibilità di vedere in anteprima mondiale questo materiale, ci siamo impegnati a non rivelare nulla di quanto ci è stato mostrato. Possiamo dirvi soltanto che le novità attese per quest'anno saranno certamente le più clamorose dall'introduzione dell'Amiga. La conferenza di Parigi non ha lasciato dubbi sulla serietà e l'energia con cui la Commodore punta al mercato professionale.

Un incredibile simulatore

Le novità non sono mancate neanche dal fronte dei produttori. La Mercuriel, un'azienda francese, ha presentato il Simobile, un simulatore di automobile ideato da un progettista che ha al suo attivo simulatori impiegati nell'industria aerospaziale. Il sistema è basato su un Amiga 2000 con videodisco. Il posto di guida, il cruscotto, i pedali, perfino la

cintura di sicurezza sono uguali a quelli di un'automobile vera. Uno schermo a colori emula nei minimi dettagli la visuale dal posto di guida, compreso lo specchietto retrovisore. Contrariamente ad altri simulatori basati sui videodischi, la Simobile non ha percorsi obbligati: le diverse strade che si possono imboccare agli incroci sono state tutte memorizzate sul disco. Inoltre, il paesaggio è stato filmato in Cinemascope da un mezzo in movimento, per cui il sistema ha sempre a disposizione le immagini di un arco più ampio di quello visualizzato normalmente, ed è pronto a servirsene se l'autista decide di buttarsi fuori strada. Abbiamo provato a spegnere il motore percorrendo una strada in leggera salita: il mezzo ha rallentato, si è fermato e ha iniziato una sempre più veloce discesa a marcia indietro! Il seggiolino viene inclinato avanti o indietro dal computer per dare un maggiore realismo alle fasi di accelerazione e frenata. Qualcuno ha chiesto dov'era il pulsante Fire, ma la Simobile non è un gioco. Con un prezzo che non supera i 25 milioni di lire, è un formidabile ausilio per le scuole guida. In Francia è già stato approvato dalle autorità competenti. L'Amiga ha reso possibile la creazione del primo di quella che presto diventerà una serie di "microsimulatori", una novità assoluta che farà nascere dal nulla un intero mercato.

Travestirsi da Atari

Qualcuno si ricorda del fantomatico emulatore Atari, annunciato qualche anno fa? Si trattava di uno scherzo, una semplice schermata senza alcun programma. Lo scherzo è ora diventato realtà. L'idea è nata proprio da quel finto emulatore. Del resto, se l'Amiga è in grado di simulare un'automobile... H.S., giovane programmatore con una lunga esperienza sull'Atari ST, ci ha fatto provare in anteprima assoluta l'emulatore Atari per l'Amiga. L'autore ci ha chiesto che non venisse pubblicato il suo nome, dal momento che si sta preparando ad affrontare una possibile battaglia legale contro la Atari, la cui reazione certamente non si farà attendere.

Sopra: il simulatore Simobile. Sotto: una suggestiva immagine visualizzata grazie a un prototipo di scheda video

È quasi certo che la soluzione per evitare problemi circa i diritti di copyright della Atari consisterà nell'utilizzare il disco originale contenente il sistema operativo dell'Atari ST. L'emulatore, infatti, è in grado di leggere e scrivere dischi Atari utilizzando i disk drive dell'Amiga. Le prestazioni sono incredibili. Abbiamo lavorato a lungo con il programma *Calligrapher*, senza incontrare il minimo problema. Anche la velocità di esecuzione, su un Amiga 2000 in configurazione standard, era più che accettabile. Secondo l'autore, che ha provato l'emulatore su un Amiga con un microprocessore 68030, in quella configurazione i programmi funzionano più velocemente che su qualsiasi modello Atari. Il motivo per cui è possibile emulare così bene l'Atari ST, ci ha spiegato l'autore, è che l'Atari è un calcolatore "vuoto", che non sfrutta alcuna caratteristica hardware esclusiva. Il microprocessore utilizzato è lo stesso dell'Amiga, mentre mancano tutti quei chip custom che rendono l'Amiga così unico e "inemulabile".

La versione finale dell'emulatore sarà probabilmente dotata di un dispositivo in grado di garantire la compatibilità hardware con alcune periferiche Atari, a cominciare dalla porta MIDI. Lo stesso dispositivo fungerà da "dongle", cioè protezione contro la copiatura, per evitare che qualcuno... emuli l'emulatore.

Amiga Loads Faster

Altra interessantissima novità è stata la nuova linea di prodotti ALF (da "Amiga Loads Faster"), presentata da Oliver Kastl della ditta tedesca Elaborate Bytes. La linea ALF copre i settori hardware e software, proponendo controller per hard disk compatibili con gli standard ST412/ST506 o SCSI, e un insieme di programmi di supporto, driver e altre preziose utility. La caratteristica più interessante della linea ALF è la possibilità di utilizzare le periferiche standard dei PC, notoriamente più economiche delle versioni per Amiga, a velocità decisamente superiori alla media.

Il nuovo sistema, denominato ALF2, permette di controllare le più svariate unità di memoria, come CD-ROM, dischi rimovibili e tape streamer standard SCSI, comodissimi per le copie di sicurez-

za. ALF2 è in grado di associare una password di accesso a ogni partizione, o d'inibire la scrittura di un supporto, proteggendo così i dati da virus e altri accessi indesiderati. Per sfruttare al massimo le possibilità del computer, ALF2 riconosce automaticamente il tipo di microprocessore utilizzato, e genera il codice più opportuno per il modulo di trasferimento dati.

Il software di ALF2 è basato su un rivoluzionario concetto di modularità e indipendenza dall'hardware. Gli stessi programmi sono in grado di funzionare indipendentemente con un hard disk, una cartuccia a nastro o una memoria ottica. È assicurata la compatibilità con una decina di controller di diverse marche, tra cui la serie 2090 della Commodore. Il programma d'installazione di un nuovo supporto di memoria utilizza l'interfaccia grafica dell'Amiga. Non è necessario né accedere alla complicata MountList Amiga, né modificare la startup-sequence. Sofisticati programmi di verifica permettono d'individuare gli errori meno prevedibili, da un byte difettoso su un disco, a un cavo collegato male. ALF2 varrebbe il suo prezzo anche soltanto per il suo software di backup, che consente le operazioni più disparate: la copia da un supporto rimovibile ad alta capacità a un altro (con un solo disk drive a disposizione), la copia da disco rigido a streamer, e viceversa, senza trascurare naturalmente i dischi standard da 880K. La scheda di ALF2 è disponibile in diverse versioni, a seconda del modello di computer, della presenza o assenza dell'autoboot (anche da Fast Filing System) e a seconda dell'interfaccia scelta (ST412, ST506 oppure SCSI). Un'eventuale scheda Janus può accedere ad ALF2 anche per effettuare il boot iniziale.

Gran finale

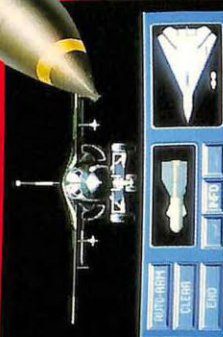
Parigi è una città dalle mille tentazioni, ma alcune persone sembrano esserne immuni. Fedeli allo stereotipo che vuole il programmatore incollato alla tastiera fino all'alba, gran parte dei partecipanti non sono mai usciti dall'albergo. Altri, invece, hanno preferito esplorare il quartiere latino, o gustare ostriche e lumache nei migliori ristoranti.

Sopra: il team Commodore artefice del software sistema dell'Amiga. Sotto: un Amiga 2000 con il sistema ALF2

Gli organizzatori della conferenza hanno comunque avuto la felice idea d'invitare tutti a una serata in maschera nell'albergo, battezzata "Napoleon's Night", con tanto di premiazione per le maschere più pazze. I partecipanti piazzatisi tra il secondo e il sesto posto hanno ricevuto in premio dalla Commodore una copia di *C1-Text*, il word processor realizzato in Italia. Tra i vincitori dell'ambito premio, Gail P. Wellington, general manager Commodore, che indossava gli abiti del duca omonimo. Il primo premio è stato invece assegnato all'unica persona che non aveva bisogno del pacchetto della Cloanto, cioè il suo autore. La maschera più pazza è stata quindi considerata "Sindrome Amiga": un avvertimento a tutti i programmatori che si espongono per troppe ore alle misteriose radiazioni emesse dal chip Fat Agnus. Con capelli verdi, pelle argentea e una misteriosissima luce rossa lampeggiante proveniente dal profondo delle orecchie, i soggetti colpiti danno evidenti segni di Guru Meditation. Sembra che si nutrano esclusivamente di topi (mouse), con una particolare predilezione per i modelli a tre pulsanti. Che sia in arrivo una nuova generazione dei temutissimi "Visitors"? Il pubblico non sembrava affatto spaventato. Un trionfale applauso è scoppiato quando il microfono del podio è stato avvicinato al premiato. L'inconfondibile voce sintetizzata dell'Amiga continuava a ripetere: «This is Amiga speaking... The Amiga Syndrome... Only Amiga makes it possible!». Senza dubbio una brillante parodia del programmatore, che ha fatto vincere al nostro Paese questa gara di fantasia e creatività.

Dopo tre giorni d'intensi lavori, anche la conferenza di Parigi è giunta al termine. Il prossimo appuntamento è con la developers conference americana, ed è stato fissato dal 27 al 29 giugno, ad Atlanta.

Dopo la conclusione della conferenza di Parigi, abbiamo saputo da Gunda O'Neal, a cui fanno capo l'organizzazione e il coordinamento delle conferenze europee, che tra le città attualmente prese in considerazione per ospitare la conferenza del 1991, Milano è quella che raccoglie più consensi. C'è quindi la possibilità di vedere l'intera comunità Amiga nel nostro Paese! ■

BOMBER
TACTICAL AIR COMBAT
Veloce e divertente da giocare, Bomber ti regala una dettagliata e accurata combinazione di simulazione di volo, combattimento aria-aria e combattimento terra-aria, per il più completo gioco di simulazione mai realizzato.
• Realistico paesaggio in 3D
• Scegli fra 7 diversi aerei da combattimento
• Ricerca e seleziona le armi
• Unico sistema di controllo della visuale esterna che permette il movimento su tutti gli assi.
MANUALE INTEGRALE IN ITALIANO!
ACTIVISION
© ACTIVISION UK LTD 1989
ALL RIGHTS RESERVED
LEADER
DISTRIBUZIONE
Amiga screen shots shown
AUTO-ARM
CLEAR
END
INFO
Mig-27
Flogger
3D
INFO
SELECT
F4 Phantom
2D
INFO
SELECT

NOVITÀ USA: DA DISNEYWORLD AI CAMPIONATORI

Inizia con questo numero una nuova serie di articoli che rappresentano la nostra finestra aperta sul mondo Commodore negli States: nuovi prodotti, anteprime, notizie, curiosità

di Morton A. Kevelson

Essendo appena tornato da un viaggio di una settimana che ha toccato Disneyworld, Cape Canaveral e le spiagge atlantiche nei pressi di Orlando, in Florida, mi scopro in preda a sintomi contraddittori: da una parte l'acuta crisi di astinenza da computer, dall'altra la nostalgia per il clima più dolce del sud. Soltanto dopo essermi reso conto di aver già speso quasi metà delle mie ferie e di essere drammaticamente in rosso con la mia carta VISA, mi sono deciso a rimettermi davanti alla tastiera.

Uno dei momenti più illuminanti del mio soggiorno a Orlando l'ho trascorso nel nuovo parco a tema della Disney, dedicato alla MGM. Reena, il più giovane membro della comitiva (ovvero una bambina di sette anni con una tenacia da non credere) era appena riuscita a trascinarci tutti e cinque in primissima fila all'epico e rocambolesco spettacolo di Indiana Jones. E nella sala di controllo, tra una lunga fila di pannelli e cinque o sei tecnici, ho intravisto le sagome non di uno ma addirittura di due Amiga 2000. In realtà, devo confessare che sulle prime quello che aveva attirato la mia attenzione erano state le palline variopinte che rimbalzavano su uno dei due schermi. I due computer venivano utilizzati per controllare tutti gli effetti sonori dello show. Erano normalissimi Amiga 2000, con 1 megabyte di memoria RAM e due disk drive. Non avevano l'hard disk, anche se il tecnico con il quale ho parlato mi ha detto che contavano di montarlo al più presto. Le macchine erano equipaggiate con una scheda d'interfaccia personalizzata e un software prodotto dalla Richmond Sound con sede a Vancouver (B. C. Canada, telefono: 001/604/732-1234). La scheda consentiva al sistema di ricevere i segnali di temporizzazione dalle apparecchiature sul set e di usarli per sincronizzare gli effetti sonori. Abbastanza stranamente, la scelta era caduta sull'Amiga non per le sue quattro voci per la produzione di suoni campionati, ma per le sue capacità multitasking. In effetti le capacità sonore interne dell'Amiga non vengono usate quasi mai per applicazioni di questo tipo. Il multitasking in tempo reale dell'Amiga, invece, lo rende l'unico computer in grado di gestire la difficile sincronizzazione dei vari eventi dello spettacolo. Dei due computer, soltanto uno veniva utilizzato effettivamente a questo scopo. L'altro serviva come sistema di backup.

Al mio ritorno ho trovato una piccola novità che mi ha rattristato: la comunità dell'Amiga ha perduto un altro produttore dato che, a quanto pare, la **C Ltd** ha chiuso i battenti. La C Ltd è stata uno dei primi produttori indipendenti che si sono dedicati all'Amiga. La società risaliva in effetti ai primissimi gloriosi giorni del VIC-20 e del C-64. In origine aveva il nome di Cardco, e con quel nome aveva prodotto la Cardboard, uno chassis di espansione multislot per il VIC-20 e una serie d'interfacce per stampanti per gli 8 bit della Commodore.

C Ltd è il nome con cui la Cardco è riemersa dalla riorganizzazione interna della società, poco dopo l'introduzione sul mercato dell'Amiga. Tra i suoi prodotti più importanti, ricordiamo le schede di espansione di memoria e le interfacce SCSI per l'Amiga 1000 e 2000. Tra i suoi più recenti progetti vorrei citare il LazerXpress, un sistema di stampante laser per l'Amiga. Inoltre, la C Ltd è stata la ditta che ha aperto la strada al Virtual Page System, un sistema che permette ai programmi d'impaginazione e ai pacchetti grafici di gestire bitmap di dimensioni illimitate.

Al momento, le linee telefoni-

che della società sono disattivate e Ed Lippert non risponde al numero di casa. Ed Lippert è il direttore marketing della C Ltd ed è uno dirigenti della società fin dai tempi della Cardco. Considerando la sua storia passata, non mi sorprenderei affatto di veder ricomparire la società in qualche altra incarnazione.

Sfortunatamente, ho appena terminato di recensire un suo prodotto per l'Amiga; un nuovo scanner che si può tenere in una mano. In conseguenza di questi nuovi sviluppi, quell'articolo, scritto appositamente per *Computer Shopper*, non può più essere pubblicato.

A proposito di *Computer Shopper*, ho appena ricevuto la mia copia del numero di gennaio: una mole spaventosa di 750 fogli, formato 28 x 35 cm, zeppa di pubblicità e di articoli dedicati per la maggior parte ai computer MS-DOS. In questo numero, *Computer Shopper* ha annunciato che cesserà di trattare i cosiddetti "computer classici": macchine che comprendono numerosi computer a 8 bit, come il Coleco Adam, lo Spectrum della Sinclair, l'Atari 400/800 e i suoi successori e il Commodore 64. Si noti che, tra questi computer, soltanto il Commodore 64 è ancora in produzione.

Computer Shopper si occupa oggi di macchine MS-DOS, del Macintosh, dell'Amiga e dell'Atari ST. Questo ordine, oltre a rispecchiare quello del sommario della rivista, mostra con tutta probabilità anche la scala di priorità assegnata dalla testata ai vari computer.

Pubblicità!

La Commodore ha completato da poco negli Stati Uniti un'importante campagna promozionale in favore dell'Amiga. La campagna consisteva in una serie di spot televisivi e in sfilze di pagine di pubblicità a colori su riviste d'informazione a diffusione nazionale, quali *Time* e *Newsweek*. Gli spot televisivi portavano la firma di George Lucas ed erano realizzati molto bene, con numerosi effetti speciali (come per esempio le case fluttuanti). I protagonisti erano personaggi celebri: astronauti, divi dello sport, presentatori e uomini politici. La pubblicità sulle riviste mostrava volti famosi e persone comuni, tutti al lavoro sul loro Amiga.

Posso testimoniare personalmente che la campagna ha avuto un buon successo, dato che mi è capitato d'incontrare molte persone che non conoscevano l'Amiga e che ora almeno sanno della sua esistenza. È ovvio però che il punto chiave si riduce alla domanda: quanti Amiga in più ha fatto vedere la campagna pubblicitaria?

Fonti ufficiose parlano di un milione e mezzo di unità vendute in tutto il mondo, la maggior parte delle quali al di fuori degli Stati Uniti. Secondo me, i problemi di diffusione dell'Amiga all'interno degli Stati Uniti sono in parte dovuti al numero insufficiente di rivenditori Amiga. Per esempio, sull'isola di Manhattan, a New York, c'è un numero ristrettissimo di negozi che dispongono dell'intera linea Amiga.

La Commodore ha recentemente dato l'avvio a una campagna per promuovere l'Amiga come computer multimediale e, in particolare, come workstation per il desktop video. Questa informazione non è certo una sorpresa per l'industria statunitense del video, un settore nel quale l'Amiga è da qualche anno uno dei protagonisti principali. Con ogni probabilità, è stato proprio il successo ottenuto dall'Amiga in questo mercato, che ha indotto la Apple a una massiccia opera di promozione del Macintosh per le stesse applicazioni. La Apple ha anche distribuito una videocassetta omaggio con il proposito di dimostrare le potenzialità del Macintosh nell'ambito del desktop. Sono riuscito a procurarmi una copia di quella videocassetta e ho constatato che, allo stadio attuale, gli effetti del Macintosh sono addirittura patetici in confronto a ciò che è in grado di fare l'Amiga. La videocassetta è piena di iperboli pubblicitarie, ma fornisce ben pochi esempi d'immagini realizzate con l'ausilio del computer.

Computer in mostra

Sul fronte delle mostre, sono stati da poco resi pubblici i dati ufficiali delle presenze al primo AmiEXPO europeo. Durante i tre giorni della mostra di Colonia, tenutasi nello scorso novembre, sono intervenuti ben 34.916 visitatori. Il sorprendente successo ha incoraggiato gli organizzatori a portare quest'anno a quattro i giorni della mostra, che si terrà dall'8 all'11 novembre nel complesso espositivo di Colonia. Il giorno 8 sarà destinato interamente all'industria, e i battenti dell'esposizione AMIGA '90 saranno aperti solo per la stampa e i produttori del settore. Negli Stati Uniti, l'AmiEXPO è stata invece fissata per il primo fine settimana di marzo e si terrà a Washington anziché a New York. La scelta di Washington ha già suscitato un notevole interesse da parte dell'industria, dato che si tratterebbe del primo Commodore Show a svolgersi nella capitale.

La mancanza di un'esposizione dedicata all'Amiga a New York nel primo semestre di quest'anno non è passata inosservata. L'Hunter Group, la società di Toronto responsabile di tutti i World of Commodore, ha annunciato per l'ultimo weekend di aprile una mostra di prodotti per l'Amiga, che si terrà proprio a New York. Conto di essere presente a entrambi gli appuntamenti in modo da potervi dare le informazioni essenziali e le ultime novità.

Hardware e ultimi arrivi

Di recente, molti nuovi prodotti hanno attirato la mia attenzione. La **Interactive Video Systems** (Garden Grove, California, telefono: 001/714/898-0858) ha presentato ufficialmente una sua versione del Disk Manager Mac,

da utilizzare con il controller per disco fisso TrumpCard e l'emulatore Macintosh per l'Amiga, A-Max (si veda la mia recensione di A-Max sul numero 4/89 della rivista). Come ho già avuto modo di sottolineare, A-Max fornisce un'eccellente emulazione del Macintosh sull'Amiga. Il difetto maggiore è la completa mancanza di qualsiasi supporto per l'hard disk Amiga. La Interactive Video Systems ha apportato alcune modifiche al software del controller TrumpCard perché funzioni con la versione modificata del Disk Manager Mac (si tratta di un programma di gestione dell'hard disk per il Macintosh).

Il pacchetto della Interactive Video Systems permette la coesistenza di partizioni Amiga e Macintosh (nelle proporzioni che si preferiscono) nello stesso hard disk. La IVS dichiara addirittura che la partizione del Macintosh è costruita in modo che l'hard disk potrebbe essere installato in un Macintosh e impiegato per il boot della macchina. Non ho potuto controllare questa notizia sensazionale, ma in questo momento ho una partizione Macintosh da 7 MB sul mio Amiga 2000, e non mi dà alcun problema. Tutti i miei complimenti alla Interactive Video Systems.

Per le applicazioni di desktop publishing, la **Sharp Electronic Corporation** insieme con la **ASDG**, ha messo in commercio il suo Handy Color Scanner, modello JX-100, per i computer Amiga. Si tratta di un'unità portatile, un tantino troppo grande perché si possa utilizzarla con una mano sola, che può effettuare la scansione su un'area di 10 centimetri per 16 con una risoluzione di 80 punti per centimetro. È altresì capace di creare immagini digitalizzate in numerosissimi formati che vanno da 1 a 18 bit per pixel. Con il primo formato si ottiene un'immagine in bianco e nero molto contrastata, mentre l'ultima genera un'immagine a colori con una palette di 262.144 possibili colori. Il software dà supporto al nuovo formato IFF a 24 bit.

Il software della ASDG dà supporto anche ai modi grafici AHAM e A Res. Questi modi grafici, insieme ai modi grafici Dynamic supportati da *DigiView 4.0* della **NewTek**, ridefiniscono la palette dell'immagine per ogni linea di scansione. Questo consente a tutti i 4096 colori di comparire sullo schermo in immagini ad alta risoluzione. Il pacchetto *ScanLab/100* sarà immesso sul mercato statunitense dalla Sharp Electronic Corporation e su quello europeo, dalla ASDG. Il software supporta i modi video NTSC e PAL in tutti i modi grafici dell'Amiga, compreso l'overscan severe.

Space Ace della **ReadySoft**: questo gioco in quattro dischi, che mette in scena animazioni originali di Don Bluth, sembra addirittura più impressionante di *Dragon's Lair*, il primo programma di questo genere prodotto dalla ReadySoft. E dà l'impressione di essere frustrante in tutto e per tutto, come il suo predecessore (si veda la recensione a pagina 30 di questo stesso numero, *n.d.r.*).

Il campionatore audio *Perfect Sound 3.0* prodotto dalla **SunRize**: il pacchetto presenta, rispetto alla versione precedente, miglioramenti sia nell'hardware che nel software. Il software ha una nuova interfaccia utente con capacità di suono e di edit aggiuntive. L'hardware dà una distorsione più bassa, dispone di un input monofonico a livello di microfono e di input stereofonico con controllo digitale nel collegamento hardware diretto. Il nuovo software funziona anche con il vecchio hardware. Gli utenti europei possono aggiornare il pacchetto restituendo l'hardware originale, il disco e il manuale, e versando una somma aggiuntiva di 50 dollari.

Gli 8 bit Commodore

Al momento in cui scrivo, la Commodore ha cessato ufficialmente anche la produzione del computer C-128D. Il vecchio C-128, che era stato rimpiazzato dal C-128D, era già fuori produzione da un po' di tempo. Insomma, la Commodore non è proprio riuscita a trovare una nicchia di mercato per questa macchina, sulla piazza statunitense. Sospetto che il problema sia dovuto alla struttura dei prezzi dei vari computer Commodore.

Un C-128D o un C-128 con un disk drive 1571 avevano un prezzo al pubblico che oscillava fra i 450 e i 500 dollari. Questo li poneva appena sotto la soglia dei 600 dollari, il prezzo minimo dei cloni XT MS-DOS. A peggiorare il problema, c'era il fatto che si poteva acquistare un Amiga 500 per circa 550 dollari. Questo poneva la Commodore nell'imbarazzante situazione di farsi concorrenza da sola.

Negli Stati Uniti, il supporto alle macchine a 8 bit sembra diminuire sempre più. La maggior parte del software che viene oggi immesso sul mercato sono videogiochi per il C-64. Ciò nonostante, vi sono ancora alcune ditte che forniscono un adeguato supporto sia al C-128 che al C-64.

La principale è la **Software Support International** (Vancouver, Washington, telefono: 001/206/695-9648). Questa società ha da poco presentato la versione 5.0 della cartuccia Super Snapshot, un'utility per il C-64/128. Alcuni altri loro prodotti sono il Toolkit 1581 di cui è appena apparsa la versione 2 e il Fastrac 128, un'utility per il C-128 che serve essenzialmente per copiare i dischi e gestire i file.

In generale, i programmi sviluppati dalla Software Support International sfruttano appieno le potenzialità delle macchine per le quali vengono scritti. Comprendono il supporto per i moduli di espansione di memoria della serie 1700 e utilizzano anche i 64K di RAM video del C-128D.

Mi piacerebbe che i prossimi articoli di questa serie rispondessero il più possibile alle vostre curiosità. Vi prego di non esitare a inviare alla rivista i vostri commenti oppure richieste di argomenti particolari. ■

PROGRAMMARE L'AMIGA

LE FUNZIONI DELLA LIBRERIA ARP

Un completo elenco delle funzioni della libreria ARP, che permette di dare uno sguardo d'insieme alle sue caratteristiche fondamentali e di valutarne la compatibilità con la libreria DOS

di Eugene P. Mortimore

Nel numero 1/89 di *Commodore Gazette* abbiamo parlato dei comandi dell'AmigaDOS Replacement Project (ARP, progetto di sostituzione dell'AmigaDOS), creati da un altruistico gruppo di programmatori consapevoli delle molte carenze del vecchio sistema.

I miglioramenti sono: comandi più agili e compatti (che risparmiano spazio su disco); la possibilità di renderli residenti (accelerando quindi i tempi di reazione della macchina); una sintassi linea comando migliore e più coerente; messaggi d'aiuto più esplicativi, e parecchi altri utili particolari.

È ovvio che, per far coesistere comandi vecchi e nuovi, i programmi in linguaggio C e Assembly che vi fanno ricorso devono garantire la "compatibilità verso il basso". Cioè devono funzionare anche con i normali comandi dell'AmigaDOS e il loro uso sulla linea comando dev'essere trasparente all'utente. Devono inoltre essere compatibili con ogni versione del sistema operativo, e in particolare con le varie versioni della libreria DOS.

L'obiettivo finale, evidentemente, è quello di sostituire la libreria DOS con una più ampia "libreria ARP". D'altra parte, come abbiamo già osservato in precedenza, la ROM dell'Amiga è talmente stipata che questo cambiamento richiederebbe una ROM più estesa.

Ricordiamo che i comandi ARP sono disponibili sul disco *TxEd* come omaggio, e si trovano anche come software di pubblico dominio.

Come abbiamo già visto, c'è una corrispondenza quasi 1:1 tra i comandi originali e quelli ARP. Per ogni comando dell'AmigaDOS, infatti, esiste un omonimo comando ARP. Non tutti i comandi, però, sono stati riscritti (i programmatori sono ancora al lavoro) e per quanto riguarda l'idea di sostituire addirittura l'AmigaDOS in ROM siamo ancora lontani: la fabbricazione di nuove ROM più capaci richiede la benedizione e l'appoggio della Commodore, e per il momento una simile soluzione è una possibilità molto vaga. Comunque la Commodore è interessata, e con un po' di pazienza prima o poi scopriremo se i comandi ARP saranno adottati ufficialmente, com'è accaduto per il sistema *AREXX* (anch'esso elaborato da programmatori indipendenti).

Ci sono due operazioni preliminari da svolgere per fare uso dei comandi ARP: 1) si devono collocare tutti i comandi nella directory C (in genere, DF0:C); 2) si deve collocare il file arp.library nella directory LIBS: (in genere, DF0:LIBS).

Fatto ciò, quando si impartisce un comando della libreria ARP – per esempio COPY – il comando viene caricato in memoria e provvede lui stesso a chiamare la funzione OpenLibrary per aprire la libreria ARP, aspettandosi di trovarla nella directory LIBS:. Se non è così, il comando invia un messaggio d'errore (per esempio "No ARP Library") segnalando che non riesce a trovare il file arp.library.

Quanto detto finora, comunque, era diretto soprattutto agli utenti, mentre noi vorremmo rivolgerci ai programmatori, spostandoci a un livello più tecnico nell'esame dei nuovi comandi. Presenteremo cioè una descrizione dell'architettura interna della libreria ARP, illustrandone le funzioni e mettendone in luce le caratteristiche principali e i motivi per cui sono state create.

Nel prossimo articolo, e nei successivi, presenteremo una descrizione dettagliata di tutte le funzioni della libreria ARP, spiegandone la sintassi, gli argomenti, i valori restituiti, e altri importan-

ti dettagli che i programmatori devono conoscere per fare di queste funzioni l'uso più appropriato. In seguito, vedremo come usare queste funzioni nell'ambito di programmi in C o in Assembly, basandoci sempre su programmi d'esempio.

Anticipiamo che in questo articolo non spiegheremo nei dettagli tutte le strutture in linguaggio C che verranno citate; per una spiegazione completa vi rimandiamo ai prossimi articoli sull'argomento.

Le funzioni della libreria ARP

Le Tavole 1 e 2 elencano tutte le funzioni contenute nella più recente versione della libreria ARP (versione 1.2). I numeri della prima colonna permettono di determinare l'offset di ciascuna funzione dalla base della libreria (utile per interpretare l'output di un disassemblatore, come abbiamo spiegato nell'articolo sul disassemblatore *DSM*, su *Commodore Gazette* numero 1/90).

Abbiamo arbitrariamente spezzato in due la libreria, che in totale contiene 102 funzioni: le prime 33 (quelle che compaiono nella prima tavola) sono esattamente le 33 funzioni della libreria DOS. In altre parole, la libreria ARP contiene come sottoinsieme la libreria DOS e dal momento che la contiene proprio all'inizio (e le funzioni sono nello stesso ordine) anche gli offset dalla base della libreria sono gli stessi. Di conseguenza, per aggiornare un vecchio programma che non faceva uso della libreria ARP è sufficiente sostituire la chiamata a OpenLibrary per aprire la libreria DOS con un'analoga chiamata a OpenLibrary diretta ad aprire la libreria ARP.

Si rammenti che in genere i programmi in C ottengono automaticamente l'apertura della libreria DOS tramite il codice di startup fornito con il compilatore (ad esempio c.o per il *Lattice C*). In altre parole, quasi nessun programma apre esplicitamente la libreria DOS e quindi nella maggior parte dei casi basta *aggiungere* una chiamata a OpenLibrary per aprire la libreria ARP. Si possono poi chiamare le funzioni elencate nella Tavola 2, senza preoccuparsi della presenza o dell'assenza della libreria DOS. È evidente che questa tecnica "guarda avanti" al momento in cui le funzioni della libreria ARP saranno finalmente in ROM.

Il comportamento con i vecchi programmi

Un programma per la gestione dei file su disco scritto prima dell'avvento della libreria ARP deve

TAVOLA 1: Le funzioni della libreria DOS

Numero interno e offset	Nome	Descrizione
1 (– 30)	Open	Apre un file
2 (– 36)	Close	Chiude un file
3 (– 42)	Read	Legge caratteri da un file
4 (– 48)	Write	Scrive caratteri in un file
5 (– 54)	Input	Identifica l'handle del file di input del programma
6 (– 60)	Output	Identifica l'handle del file di output del programma
7 (– 66)	Seek	Si sposta a una specifica posizione in un file
8 (– 72)	DeleteFile	Cancella un file
9 (– 78)	Rename	Cambia nome a un file
10 (– 84)	Lock	Chiede un accesso in lettura o in scrittura a una directory o a un file
11 (– 90)	UnLock	Rimuove un lock da una directory o da un file
12 (– 96)	DupLock	Duplica un lock di lettura condiviso di una directory o di un file
13 (– 102)	Examine	Esamina un file o una directory associati a un lock
14 (– 108)	ExNext	Esamina la voce successiva della directory
15 (– 114)	Info	Ottiene informazioni sul contenuto di un disco
16 (– 120)	CreateDir	Crea una nuova directory
17 (– 126)	CurrentDir	Trasforma una directory nella directory di lavoro
18 (– 132)	IoErr	Restituisce informazioni ulteriori quando una funzione della libreria restituisce un codice d'errore
19 (– 138)	CreateProc	Crea un nuovo processo
20 (– 144)	Exit	Esce dal programma utilizzando il codice di ritorno indicato
21 (– 150)	LoadSeg	Carica in memoria un modulo di programma
22 (– 156)	UnLoadSeg	Rimuove dalla memoria un modulo di programma
23 (– 162)	GetPacket	Ottiene una struttura Packet DOS per l'esame del programma
24 (– 168)	QueuePacket	Accoda una struttura Packet DOS nella message port di un task
25 (– 174)	DeviceProc	Restituisce l'identificatore di un processo che gestisce l'I/O
26 (– 180)	SetComment	Aggiunge un commento a un file
27 (– 186)	SetProtection	Imposta i bit di protezione per un file o una directory
28 (– 192)	DateStamp	Ottiene data e ora nel formato standard dell'AmigaDOS
29 (– 198)	Delay	Sospende l'esecuzione di un programma per un indicato numero di tick
30 (– 204)	WaitForChar	Indica se entro un certo limite di tempo arriva un carattere
31 (– 210)	ParentDir	Ottiene la directory genitore di un file o di una directory
32 (– 216)	IsInteractive	Scopre se un file è collegato a un terminale virtuale
33 (– 222)	Execute	Esegue un comando CLI dall'interno del programma

come minimo: 1) aprire la libreria DOS (usando direttamente una chiamata a OpenLibrary oppure contando sull'apertura automatica eseguita dal codice di startup fornito con il compilatore); 2) aprire un file tramite la funzione Open della libreria DOS; 3) leggere un certo numero di caratteri da quel file tramite la funzione Read della libreria DOS; 4) scrivere in quel file un certo numero di caratteri tramite la funzione Write della libreria DOS; 5) chiudere il file tramite la funzione Close della libreria DOS.

Allo stesso modo, i programmi che avevano a che fare con le directory utilizzavano, tra l'altro, le seguenti funzioni della libreria DOS: ParentDir, CurrentDir, CreateDir, Rename, Lock, UnLock, DupLock, ExNext, Examine, Info, IoErr.

I programmi che trasferivano in buffer personali i caratteri provenienti dalla tastiera (per esempio per trasferirli in seguito nei file destinazione tramite la funzione Write) ricorrevano alle funzioni Input, Output, IsInteractive e WaitForChar.

I programmi che volevano mandare in esecuzione al loro interno un programma subordinato (processo o task), e in seguito riprendere l'esecuzione dal punto in cui si erano interrotti, in genere dovevano ricorrere alle funzioni LoadSeg, Execute e UnLoadSeg.

I programmi che dovevano trattare con i processi (simili ai task) facevano uso delle funzioni CreateProc e DeviceProc.

I programmi che dovevano trattare con pacchetti DOS facevano uso delle funzioni GetPacket e QueuePacket.

La filosofia generale è la seguente: anche se un programma ha bisogno soltanto delle *nuove* funzioni della libreria ARP (quelle della Tavola 2), dal momento in cui la apre è comunque in grado di avere accesso alle funzioni della libreria DOS. La libreria ARP è completamente indipendente.

Le nuove funzioni

Prendiamo ora in esame le nuove funzioni, per cercare di capirne lo scopo e la filosofia. Abbiamo articolato la nostra analisi scomponendo la Tavola 2 in alcuni sottoinsiemi.

1. C'è un gruppo di funzioni che sono semplicemente utili estensioni delle normali funzioni delle librerie linked in linguaggio C. In linea generale,

TAVOLA 2: Le nuove funzioni della libreria ARP

Numero interno e offset	Nome	Descrizione
34 (– 228)	Printf	Stampa dati formattati nel flusso di output
35 (– 234)	FPrintf	Stampa dati formattati in un file su disco
36 (– 240)	Puts	Stampa una stringa con un carattere New Line nel flusso di output
37 (– 246)	ReadLine	Intercetta una riga di caratteri dal flusso di input
38 (– 252)	GADS	Analizza gli argomenti della linea comando dell'AmigaDOS
39 (– 258)	Atol	Converte una stringa ASCII in un intero di tipo LONG
40 (– 264)	EscapeString	Converte i caratteri Escape in una stringa linea comando
41 (– 270)	CheckAbort	Controlla se l'utente ha digitato la combinazione Ctrl/C
42 (– 276)	CheckBreak	Controlla se l'utente ha inviato un segnale di break (Ctrl/C-D-E-F)
43 (– 282)	Getenv	Ottiene il valore di una variabile di ambiente
44 (– 288)	Setenv	Imposta il valore di una variabile di ambiente
45 (– 294)	FileRequest	Apre un requester che permette all'utente di selezionare i file
46 (– 300)	CloseWindowSafely	Chiude una finestra di Intuition dopo aver restituito tutti i messaggi IDCMP
47 (– 306)	CreatePort	Crea una message port in stile Exec
48 (– 312)	DeletePort	Cancella una message port in stile Exec
49 (– 318)	SendPacket	Invia una struttura Packet DOS a un packet handler
50 (– 324)	InitStdPacket	Inizializza una struttura Packet DOS standard
51 (– 330)	PathName	Costruisce il completo percorso di ricerca di un file o di una directory
52 (– 336)	Assign	Assegna il nome di un dispositivo logico a un dispositivo fisico
53 (– 342)	DosAllocMem	Alloca un blocco di memoria in un modo compatibile all'AmigaDOS
54 (– 348)	DosFreeMem	Libera un blocco di memoria precedentemente allocato
55 (– 354)	BtoCStr	Converte una stringa BSTR in una stringa a terminazione nulla
56 (– 360)	CtoBStr	Converte una stringa a terminazione nulla in una stringa BSTR
57 (– 366)	GetDevInfo	Permette al programma di attraversare una lista di strutture DevInfo
58 (– 372)	FreeTaskResList	Libera tutte le risorse software di cui un task mantiene il track
59 (– 378)	ArpExit	Esce dal programma chiudendo la libreria ARP e liberando tutte le risorse di cui è mantenuto il track
60 (– 384)	ArpAlloc	Alloca un blocco di memoria e ne mantiene il track
61 (– 390)	ArpAllocMem	Alloca un blocco di memoria del tipo richiesto e ne mantiene il track
62 (– 396)	ArpOpen	Apre un file AmigaDOS e gli fornisce un meccanismo di track
63 (– 402)	ArpDupLoc	Duplica il lock di un file o di una directory e ne mantiene il track
64 (– 408)	ArpLock	Ottiene un lock per un file o una directory e ne mantiene il track
65 (– 414)	RListAlloc	Alloca memoria per una specifica lista di risorse e ne mantiene il track

- SEGUE -

- *SEGUE* -

66	(– 420)	FindCLI	Fornisce un puntatore a una struttura Process collegata a uno specifico processo CLI
67	(– 426)	QSort	Ordina rapidamente un array utilizzando l'algoritmo di Hoare
68	(– 432)	PatternMatch	Confronta fra loro una stringa di testo e una stringa con caratteri jolly
69	(– 438)	FindFirst	Cerca un file di cui si conosca il nome
70	(– 444)	FindNext	Cerca il file successivo al primo che si accorda a una certa stringa di ricerca
71	(– 450)	FreeAnchorChain	Disalloca una struttura AnchorPath
72	(– 456)	CompareLock	Confronta due lock di file o directory
73	(– 462)	FindTaskResList	Fornisce un puntatore alla lista di risorse del task (struttura ResList)
74	(– 468)	CreateTaskResList	Crea una nuova lista di risorse software per un task
75	(– 474)	FreeResList	Libera tutte le risorse software di cui un task sta tenendo il track
76	(– 480)	FreeTrackedItem	Libera una risorsa appartenente a una lista
77	(– 486)	GetTracker	Alloca e inizializza una struttura Tracker che si collega a una risorsa del programma
78	(– 492)	GetAccess	Rende impossibile rilasciare una risorsa di cui si sta eseguendo il track
79	(– 498)	FreeAccess	Decrementa il contatore d'uso relativo a una risorsa software
80	(– 504)	FreeDAList	Libera una struttura DAList allocata dalla funzione DosAllocMem
81	(– 510)	AddDANode	Aggiunge una nuova voce a una lista di strutture DAList
82	(– 516)	AddDADevs	Fornisce al programma una copia privata della struttura DevInfo
83	(– 522)	Strcmp	Confronta due stringhe ignorando le maiuscole
84	(– 528)	Strncmp	Confronta N caratteri di due stringhe ignorando le maiuscole
85	(– 534)	ToUpper	Converte in maiuscolo un carattere
86	(– 540)	SyncRun	Manda in esecuzione un programma separato in modo sincrono
87	(– 546)	ASyncRun	Manda in esecuzione un programma separato in modo asincrono
88	(– 552)	LoadPrg	Carica in memoria un programma cercandolo in un percorso di ricerca
89	(– 558)	PreParse	Crea una stringa token, che dovrà essere elaborata da PatternMatch
90	(– 564)	StampToStr	Converte la parte DateStamp di una struttura DateTime in una stringa leggibile
91	(– 570)	StrToStamp	Trasforma una stringa-data in formato ASCII in una struttura DateStamp dell'AmigaDOS
92	(– 576)	ObtainResidentPrg	Mette un programma residente a disposizione del programma chiamante
93	(– 582)	AddResidentPrg	Aggiunge un programma alla lista dei programmi residenti
94	(– 588)	RemResidentPrg	Rimuove un programma dalla lista dei programmi residenti
95	(– 594)	UnLoadPrg	Scarica dalla memoria un programma precedentemente caricato con LoadPrg
96	(– 600)	LMult	Esegue una moltiplicazione tra long word utilizzando interi con segno a 32 bit
97	(– 606)	LDiv	Esegue una divisione tra long word utilizzando interi con segno a 32 bit
98	(– 612)	LMod	Calcola il modulo di una long word utilizzando numeri interi a 32 bit
99	(– 618)	CheckSumPrg	Calcola il checksum di un programma residente
100	(– 624)	TackOn	Aggiunge il nome di un file a un percorso di ricerca
101	(– 630)	BaseName	Restituisce il basename di un percorso di ricerca
102	(– 636)	ReleaseResidentPrg	Decrementa il contatore d'uso di un programma residente

sono 11 le funzioni che ricadono in questa categoria. Per esempio **QSort**, che fornisce un meccanismo di ordinamento alfabetico più efficiente di quello delle funzioni contenute nelle librerie C dei compilatori *Manx* e *Lattice*. In alcune situazioni, come per esempio nell'ordinamento di array particolarmente lunghi, la maggior efficienza di QSort può essere importantissima per il programmatore.

Le altre dieci funzioni sono: **Atol**, **LDiv**, **LMod**, **LMult**, **FPrintf**, **Printf**, **Puts**, **Strcmp**, **Strncmp** e **ToUpper**. Come si può vedere esaminando i manuali dei pacchetti *Manx* o *Lattice*, queste funzioni sono molto simili alle funzioni delle librerie linked in linguaggio C, ma con alcune differenze che le rendono uniche e utilissime. Anche i nomi sono simili, se si eccettua la differenza nelle iniziali (che nel C sono minuscole).

Per esempio, sia Strcmp che Strncmp permettono di confrontare stringhe di caratteri senza far caso alle maiuscole/minuscole; LMult, LMod e LDiv, al contrario di altre funzioni più convenzionali, permettono operazioni con interi a 32 bit, diventando quindi importantissime con il bus a 32 bit delle macchine dotate di 68020/30.

In più, queste funzioni appartengono a una libreria shared, non linked, e permettono quindi a tutti i programmi che ne fanno uso di essere più agili e più efficienti.

2. Ci sono alcune funzioni che appaiono in librerie linked del C, e che sono state incluse nella libreria ARP principalmente per consentire ai programmi di utilizzarle senza doverle includere nel programma stesso. Due esempi di questo tipo sono le funzioni **CreatePort** e **DeletePort**.

Queste due funzioni sono esattamente identiche a quelle contenute nella libreria amiga.lib, che talvolta viene usata in fase di linking. Il fatto d'includerle nella libreria ARP, che è invece una libreria shared, rende possibile evitare il link della libreria amiga.lib (che rallenterebbe la fase di link e costituirebbe uno spreco di memoria).

3. Tra le funzioni a cui si poteva accedere soltanto dal CLI c'è **Assign**, che permette di assegnare un dispositivo logico a un dispositivo fisico. Con la nuova versione di Assign, ora si può

compiere questa operazione anche dall'interno di un programma in C o in Assembly.

4. Uno dei problemi con il software sistema Intuition è quello di chiudere *con tranquillità* le finestre di Intuition. Infatti, ogni finestra che possiede una message port IDCMP (si veda l'articolo "Il meccanismo IDCMP: i messaggi tra Intuition e i task", su *Commodore Gazette* numero 1/90) ha di norma un certo numero di messaggi accodati quando l'utente impartisce il comando di chiusura. Ora, se il programma non si preoccupa di restituire esplicitamente questi messaggi alla window port, il programma si blocca e la macchina va in Guru Meditation. Come dice il manuale di Intuition: "I fuochi artificiali sono assicurati". Sono molti i programmatori che hanno imparato questa lezione sulla propria pelle.

Per ovviare all'inconveniente, tra le funzioni della libreria ARP è stata inclusa la funzione **CloseWindowSafely**, che rileva automaticamente la presenza di messaggi accodati alla user port, e prima di chiudere la finestra li restituisce alle routine interne di Intuition, ovvero alla window port. CloseWindowSafely risparmia al programmatore i codici necessari per chiudere la finestra "con sicurezza": è sufficiente sostituire tutte le chiamate a CloseWindow (della libreria Intuition) con chiamate a CloseWindowSafely.

5. La libreria ARP include due funzioni – GetEnv e SetEnv – che contribuiscono ad adattare ai bisogni dei programmi l'ambiente di programmazione (o genericamente l'ambiente operativo della macchina).

Si ricordi che la libreria C della Manx offre la funzione getenv (tutte lettere minuscole), e che fu la Manx a creare il concetto di variabile ambientale e delle procedure associate. Comunque Getenv e Setenv permettono una gestione più efficace dell'ambiente operativo o di programmazione, rispetto alla funzione getenv della Manx. **Getenv** permette al programma di ottenere il valore di una variabile di ambiente e di collocarlo in un buffer di testo locale per esaminarlo ed eventualmente utilizzarlo.

Setenv permette al programma d'impostare il valore di una variabile di ambiente (da un valore precedentemente immagazzinato in un buffer locale) e di collocarlo eventualmente in un altro buffer. Si veda il manuale dell'utente della Manx, per una miglior descrizione della funzione getenv e per comprendere meglio il significato e l'uso delle variabili d'ambiente.

6. La libreria ARP fornisce due funzioni – BtoCStr e CtoBStr – che permettono al programma di trasformare normali stringhe a terminazione nulla in linguaggio C in stringhe BSTR e viceversa.

Si ricordi che il sistema operativo AmigaDOS è basato sul linguaggio BCPL, e utilizza un tipo di variabile chiamato BSTR (stringa BCPL). Un BSTR punta a una locazione di memoria che contiene nel primo byte la lunghezza della stringa e nell'altro i byte dei singoli caratteri.

Il BSTR localizza quindi la stringa in memoria e fornisce direttamente anche il conteggio dei caratteri contenuti; contrariamente alle consuete stringhe del linguaggio C, non fa riferimento a un byte NULL per indicare la fine della stringa.

Un BPTR è, invece, un puntatore diviso per quattro (in effetti punta a un blocco di memoria allineato alle LONG WORD, ovvero gruppi di quattro byte). In altre parole, l'effettivo indirizzo di memoria della stringa viene diviso per quattro prima di essere collocato nel valore BSTR.

Così, per utilizzare un BPTR o un BSTR in C, si deve semplicemente effettuare uno scorrimento a sinistra di due bit, il che significa moltiplicarlo per quattro. Non è difficile capire che questo sistema costringe a intricate linee di programma semplicemente per arrivare ai veri puntatori alle stringhe di testo.

Dal momento che le prime 33 funzioni della libreria ARP sono costrette a lavorare con i BSTR e i BPTR, con i quali lavorano anche le routine interne della libreria DOS, sono state aggiunte alla libreria ARP queste due funzioni di conversione delle stringhe, allo scopo di poter passare senza difficoltà da un tipo all'altro.

CtoBStr converte una normale stringa a terminazione nulla in una stringa BSTR e **BtoCStr** esegue l'operazione inversa.

Facendo uso di queste due funzioni, il programmatore può fare in modo che tutti gli argomenti stringa passati alle funzioni del programma (tanto le vecchie funzioni della libreria DOS quanto le nuove funzioni della libreria ARP) siano sempre corretti.

7. La libreria ARP fornisce due funzioni – StrtoStamp e StamptoStr – che permettono al programma di manipolare diversi formati di data e ora. L'originale libreria DOS contiene la funzione DateStamp che mette a disposizione del programma la struttura DateStamp nel formato interno dell'AmigaDOS.

La funzione **StrtoStamp** trasforma una stringa-data in formato ASCII in una struttura DateStamp dell'AmigaDOS. La funzione **StamptoStr** esegue il compito inverso. Le due funzioni riconoscono quattro formati di dati: il formato interno dell'AmigaDOS (dd-mmm-yy), quello internazionale (yy-mmm-dd), quello americano (mm-dd-yy) e quello canadese (dd-mm-yy).

Queste funzioni sono particolarmente utili quando si desidera un meccanismo per cambiare agevolmente il formato della data, soprattutto nel caso di un programma destinato a essere utilizzato in Paesi diversi.

8. Dalla Tavola 1 si vede che la libreria DOS fornisce due funzioni – GetPacket e QueuePacket – per gestire strutture DOS Packet. Le strutture

Packet intervengono in ogni comunicazione tra processi effettuata tramite le routine interne dell'AmigaDOS (le strutture Packet sono descritte nel *Manuale dell'AmigaDOS* pubblicato dalla IHT Gruppo Editoriale). Per la gestione dei Packet, la libreria ARP aggiunge due funzioni: InitStdPacket e SendPacket.

InitStdPacket rende automatica l'inizializzazione di una struttura Packet. Incidentalmente, ricordiamo che l'AmigaDOS fornisce venti diversi tipi di strutture Packet e permette ai programmi di specificare a quali message port dev'essere inviata una certa struttura Packet, una volta che sia stata opportunamente inizializzata.

InitStdPacket permette di automatizzare tutte le inizializzazioni delle strutture Packet, purché si faccia uso degli argomenti appropriati.

La funzione **SendPacket** invia un struttura DOS Packet a un Packet handler (si veda *Il Manuale dell'AmigaDOS*). Come con InitStdPacket, si specifica il compito del Packet e i suoi argomenti. In aggiunta, si specifica l'indirizzo del Packet handler che elaborerà il pacchetto. SendPacket è progettata in modo che il programma si metta in attesa finché l'handler non restituisce il Packet dopo averlo elaborato. In altre parole, SendPacket non può essere utilizzata per l'I/O asincrono.

9. Dalla Tavola 1 si vede che la libreria DOS non contiene nessuna funzione per l'allocazione della memoria. D'altra parte, la libreria Exec possiede numerose funzioni per gestire (allocare e rilasciare) blocchi di memoria con specifiche caratteristiche (si veda *Programmare l'Amiga Vol. I* sempre della IHT Gruppo Editoriale). La libreria ARP, comunque, fornisce quattro funzioni che vanno ad ampliare quelle della libreria Exec: DosAllocMem, DosFreeMem, ArpAlloc e ArpAllocMem.

Le prime due – DosAllocMem e DosFreeMem – sono compatibili con l'AmigaDOS. Grazie a esse, il programma può gestire blocchi di memoria di grandezza data e caratteristiche date (tra le caratteristiche, ci sono per esempio l'attivazione dei flag MEMF_PUBLIC o MEMF_CHIP). Come dicono i nomi, **DosAllocMem** alloca i blocchi di memoria e **DosFreeMem** li rilascia. Sono state incluse nella libreria ARP principalmente per essere usate internamente, al fine di costruire nodi di grandezza variabile per le liste di strutture gestite internamente dalla libreria ARP.

DosFreeMem è in grado di disallocare blocchi di memoria allocati direttamente dal programma tramite DosAllocMem, o allocati indirettamente tramite operazioni interne dell'AmigaDOS o della libreria ARP. Si osservi che se il programma vuole disallocare un blocco di memoria allocato dall'AmigaDOS, prima di chiamare DosFreeMem deve trasformare il puntatore BPTR al blocco di memoria in un normale puntatore C.

Le altre due funzioni di allocazione, ArpAlloc e ArpAllocMem lavorano in modo molto simile alle due già descritte, con qualche capacità in più. In particolare, la possibilità di tenere il conto delle allocazioni di blocchi di memoria fatte dal programma, per fare in modo che – in seguito – le routine interne della libreria ARP li restituiscano automaticamente al pool di memoria libera del sistema.

ArpAlloc alloca esplicitamente memoria Public (MEMF_PUBLIC) e Chip (MEMF_CHIP) e fornisce automaticamente un meccanismo di controllo per disallocare gli stessi blocchi quando il programma chiama le funzioni CloseLibrary o ArpExit. Come spiegheremo in uno dei prossimi articoli, le procedure seguite dalle due funzioni sono diverse se i programmi sono scritti in C o in Assembly.

La funzione **ArpAllocMem** differisce da ArpAlloc per il fatto che permette al programmatore di specificare i requisiti di memoria che si preferiscono. In realtà, ArpAllocMem è identica alla funzione AllocMem dell'Exec, se non per il fatto che si serve del meccanismo di "track" della libreria ARP per tenere il conto delle allocazioni di memoria.

10. La memoria allocata con ArpAllocMem può essere disallocata con tre funzioni: FreeTrackedItem, CloseLibrary e ArpExit. Questo ci porta a una discussione generale sulle funzioni di "track" della libreria ARP.

La libreria ARP include inoltre otto funzioni – CreateTaskResList, FreeTaskResList, FindTaskResList, RListAlloc, FreeTrackedItem, GetTracker, GetAccess e FreeAccess – che servono per tenere il conto (track) delle principali risorse del programma.

Le risorse software del programma generalmente possono essere classificate come specifici esemplari di strutture definite dal sistema o dal programma stesso. È importantissimo liberare la memoria occupata da una risorsa software non appena questa non è più necessaria.

Le otto funzioni ora elencate sono importanti perché permettono al programmatore di gestire con grande facilità (allocare, tenere il conto e rilasciare) quasi tutte le risorse software che servono in un programma.

Si può pensare a queste funzioni come a un'estensione del meccanismo di bookkeeping utilizzato dalle routine dell'Exec per tenere il conto delle strutture Library e Device aperte dai task attivi. Le funzioni di track dell'ARP utilizzano la struttura Tracker descritta nel file INCLUDE arp.h.

La funzione **CreateTaskResList** crea una nuova lista di risorse software per un task. In genere non si fa ricorso alla funzione direttamente, ma ci si appoggia all'automatico meccanismo di track della libreria ARP. In ogni caso, si può ricorrere a CreateTaskResList per creare una lista di risorse *nidificata* per un particolare task. Quando CreateTaskResList restituisce il controllo, tutte le risorse software allocate da quel task vengono raccolte in una lista *privata* che rimane attiva finché il programma non chiama la funzione FreeTaskRe-

sList.

Questo funzionamento permette, per esempio, di chiamare CreateTaskResList all'inizio di una funzione in linguaggio C e FreeTaskResList alla fine della stessa funzione. In tal caso, ogni risorsa software sfruttata all'interno della funzione viene automaticamente liberata prima che la funzione restituisca il controllo. Tutte le altre risorse di cui il programma tiene il conto, per il momento non vengono liberate.

La funzione **FreeTaskResList** libera tutte le risorse software di cui un task sta tenendo il conto. In genere FreeTaskResList viene chiamata indirettamente dalle funzioni CloseLibrary o ArpExit.

Un programma può utilizzare la funzione **FindTaskResList** per avere un puntatore alla lista di risorse del task (una struttura ResList). Se il task non sta tenendo il conto di nulla, la funzione restituisce il valore NULL.

La funzione **RListAlloc** permette al programma di allocare e tenere il conto della memoria assegnata a una specifica lista di risorse, anziché alla più recente lista a cui il programma fa riferimento. Si può fare uso di RListAlloc per eseguire il link delle risorse di memoria all'interno della lista di risorse software *privata* del programma. RListAlloc è simile a ArpAlloc, con la differenza che esegue il track per una specifica struttura ResList.

La funzione **GetTracker** alloca e inizializza una struttura Tracker, che da quel momento in avanti resta collegata a una risorsa software del programma; quando GetTracker restituisce il controllo, il programma deve provvedere a completare l'inizializzazione. Fatto ciò, le routine interne della libreria ARP eseguono automaticamente il track di quella risorsa software, incrementando o decrementando automaticamente il relativo contatore d'uso nel corso dello svolgimento del programma.

Seguire questa strada risparmia parecchie righe di programma per aprire e chiudere esplicitamente le risorse software. Si studi a questo proposito il modo in cui si comporta, per quanto riguarda allocazione e disallocazione delle risorse software, il nostro programma d'esempio sulle procedure d'azione di Intuition (pubblicato in questo stesso numero della rivista), grazie all'uso di una variabile "mask".

La funzione **FreeTrackedItem** libera una risorsa per la quale si sta eseguendo il track, e appartenente a una lista di risorse del programma. La chiamata può essere eseguita direttamente dal programma oppure automaticamente, tramite operazioni interne del meccanismo di track della libreria ARP. Quando FreeTrackedItem restituisce il controllo, la memoria occupata dalla struttura Tracker e dalla struttura della risorsa vengono restituite al pool di memoria libera del sistema.

GetAccess pone un Lock a una risorsa di cui si sta eseguendo il track (rende cioè impossibile rilasciarla); se la risorsa è già stata liberata tramite FreeAccess, la chiamata a GetAccess non ha alcun effetto.

La funzione **FreeAccess** decrementa il contatore d'uso relativo a una particolare risorsa software. Se il contatore va a -1 la risorsa viene automaticamente rilasciata dalla libreria ARP oppure da un'automatica chiamata di sistema alla funzione Expunge.

In tal modo, le routine interne della libreria ARP assicurano automaticamente che la memoria venga restituita al pool della memoria libera di sistema non appena è possibile.

Ora illustreremo brevemente i diversi tipi di risorse software di cui può essere eseguito il track in un programma.

- Innanzitutto le stesse strutture Tracker. Ogni struttura Tracker è allocata e inizializzata tramite una chiamata alla funzione GetTracker.
- I lock di file o directory creati da chiamate alle funzioni ArpLock o ArpDupLock. Il track dei lock creati con le funzioni Lock e DupLock della libreria DOS *non può* essere affidato al meccanismo automatico di track della libreria ARP. Si veda (più avanti) la descrizione del track di tipo generico.
- Le strutture FileHandle create tramite la funzione ArpOpen della libreria ARP. Il track delle strutture FileHandle create con la funzione Open della libreria DOS *non può* essere affidato al meccanismo automatico di track della libreria ARP. Si veda, anche in questo caso, la descrizione del track di tipo generico.
- Le strutture Window create da chiamate alla funzione OpenWindow di Intuition.
- Le strutture Screen create da chiamate alla funzione OpenScreen di Intuition.
- Le strutture Library della libreria Exec create da chiamate alla funzione OpenLibrary dell'Exec.
- I blocchi di memoria creati da chiamate alla funzione DosAllocMem della libreria ARP.
- I blocchi di memoria creati da chiamate alla funzione AllocEntry della libreria Exec.
- I BPTR alle strutture SegList create da chiamate alla funzione LoadSeg della libreria DOS.
- Le strutture ResList create da chiamate alla funzione RListAlloc della libreria ARP.
- I blocchi di memoria creati da chiamate alla funzione ArpAllocMem della libreria ARP.
- Le strutture DAList create da una chiamata alla funzione AddDANode della libreria ARP.
- Le strutture AnchorPath.
- Tutte le altre risorse software personalizzate che il programmatore definisce e utilizza nel programma.

11. Come mostra la Tavola 1, la libreria DOS non fornisce alcuna funzione che tratti direttamente con i dispositivi e con le relative liste di programma. La libreria ARP colma questa carenza con quattro funzioni – AddDANode, AddDADevs, FreeDAList e GetDevInfo – studiate appositamente per la gestione dei dispositivi.

La funzione **AddDANode** consente al programma di aggiungere una nuova voce a una lista DAList. Tutti i nodi creati con AddDANode sono strutture DirectoryEntry (per la descrizione di questa struttura si veda il file arpbase.h, che sarà oggetto di uno dei nostri prossimi articoli).

La funzione **AddDADevs** permette al programma di ottenere una copia privata di una lista DevInfo. Fornisce inoltre un metodo per cercare specifici tipi o voci della lista DevInfo, e trasferirli in una lista DAList privata (senza rischi per il multitasking). AddDADevs chiama automaticamente AddDANodes per aggiungere i nuovi nodi alla lista privata DAList. La lista viene ordinata prima per classi, e quindi alfanumericamente secondo i nomi dei dispositivi. Per il rilascio dei nodi allocati tramite AddDADevs è responsabile il programma.

La funzione **FreeDAList** libera una struttura DAList allocata dalla funzione DosAllocMem. Si può utilizzare FreeDAList per liberare qualunque lista di tipo DAList: l'unico requisito è che tutti i nodi della lista siano stati allocati dalla funzione DosAllocMem, e che ogni nodo contenga come prima voce un puntatore al nodo successivo. Tutti i nodi da quello indicato fino all'ultimo della lista, vengono quindi disallocati.

La funzione **GetDevInfo** permette al programma di attraversare una lista di strutture DevInfo, esaminandone ciascun nodo senza creare problemi al multitasking. Si noti che, non essendo effettuata alcuna conversione maiuscolo/minuscolo sul nome originario della struttura DevInfo, la funzione AddDADevs può fornire una tecnica di ricerca migliore.

12. La libreria ARP contiene cinque funzioni – GADS, EscapeString, ReadLine, PatternMatch e PreParse – che si possono classificare grossomodo come funzioni per la gestione della linea comando. Queste funzioni estendono le normali funzioni utilizzate internamente dall'AmigaDOS per elaborare i comandi impartiti tramite CLI.

La funzione **GADS** è studiata per analizzare (separando i caratteri in gruppi) gli argomenti standard della linea comando, facendo uso dei template standard. Si può utilizzarla per gestire linee comandi inserite interattivamente dall'utente. La funzione esegue una preelaborazione analizzando i caratteri Escape inseriti dall'utente.

Una delle ultime aggiunte alla libreria ARP permette ad alcuni dei comandi ARP di riconoscere un numero illimitato di argomenti; questa caratteristica viene chiamata "multiargs". Una simile estensione permetterà di progettare programmi capaci di gestire, per esempio, un numero illimitato di file.

La funzione **EscapeString** converte i caratteri Escape di una stringa specificata. EscapeString scandisce byte per byte la stringa specificata e sostituisce tutti i caratteri Escape con il relativo valore di byte. Attualmente la funzione riconosce i seguenti caratteri speciali:

N	carattere NewLine
T	carattere TAB orizzontale
V	carattere TAB verticale
B	carattere BackSpace
R	carattere Return
F	carattere FormFeed
E	generico carattere Escape (ASCII 27)
Xnn	ogni carattere esadecimale

La funzione **ReadLine** intercetta una riga dal flusso di input e ne colloca i caratteri in un buffer di testo mantenuto dal programma. ReadLine non permette di correggere o di visualizzare l'echo della linea comando, ma garantisce che la stringa restituita sarà a terminazione nulla. Un programma può servirsi di ReadLine per ricevere dall'utente una nuova riga di input.

La funzione **PatternMatch** sottopone a confronto una stringa con caratteri "wildcard" e una stringa di testo, determinando se la seconda si adatta allo schema di ricerca della prima. La funzione si può utilizzare per qualunque tipo di confronto fra stringhe (per esempio in una routine di ricerca all'interno di un word processor). PatternMatch dà supporto a un completo insieme di schemi di ricerca "Unix-like", e fornisce quindi una grande flessibilità nella ricerca delle stringhe.

La funzione **PreParse** crea una stringa "token" pronta a essere elaborata dalla funzione PatternMatch. PreParse parte da una normale stringa origine con caratteri jolly e crea la stringa token. Il programma usa poi la stringa token come input per la funzione PatternMatch.

13. La Tavola 1 mostra che la libreria DOS mette a disposizione molte funzioni per la gestione di file e directory. La libreria ARP le estende includendone altre 11: ArpOpen, ArpLock, ArpDupLock, CompareLock, FindFirst, FindNext, FreeAnchorChain, FileRequest, PathName, BaseName e TackOn.

Quattro di queste funzioni (ArpOpen, ArpLock, ArpDupLock e FindNext) sono molto simili alle loro controparti della libreria DOS (Open, Lock, DupLock ed ExNext). Le altre sette sono state aggiunte per facilitare ulteriormente il lavoro con file e directory.

ArpOpen apre un file AmigaDOS e gli fornisce automaticamente un meccanismo di track; quindi alloca e inizializza automaticamente una struttura Tracker.

Dal punto di vista funzionale, ArpOpen è simile alla funzione Open della libreria DOS (richiede gli stessi argomenti), con la differenza che ArpOpen, nel momento in cui il programma chiama CloseLibrary per chiudere la libreria ARP, *ricorda* l'apertura del file e lo chiude automaticamente liberando le risorse software da esso impiegate (per esempio, la sua struttura FileHandle). La stessa cosa avviene se si effettua una chiamata a FreeTrackedItem oppure a ArpExit.

Analogamente, la funzione **ArpLock** è simile alla funzione Lock della libreria DOS (richiede gli stessi argomenti) ma in più fornisce una struttura Tracker per mantenere il track della struttura Lock relativa al file o alla directory. Ricordiamo che una struttura Lock (un nome poco felice) fornisce al filing system AmigaDOS un modo per concatenare un insieme di file e per puntare alla locazione fisica del file sul disco. Anche in questo caso, grazie al track automatico, al momento della chiamata di CloseLibrary, FreeTrackedItem o ArpExit, la memoria riservata alla struttura Lock viene rilasciata. Lo stesso discorso vale per la funzione **ArpDupLock**, identica alla funzione DupLock della libreria DOS ad eccezione del track.

La funzione **CompareLock** è una novità della libreria ARP: nella libreria DOS non esiste alcuna funzione a essa paragonabile. Un programma può ricorrere a CompareLock per determinare se le strutture Lock di due file (create dalle funzioni Lock, DupLock, ArpLock o ArpDupLock) sono identiche, ovverossia appartengono allo stesso file o directory. CompareLock è stato incluso principalmente per completare la funzione Rename della libreria DOS, dal momento che impedisce di cambiare il nome di un file se quel nome esiste già in un'altra directory.

La funzione **FindFirst** può essere utilizzata per cercare specifici file, di cui sia noto il nome. La funzione individua il primo file il cui nome coincide con quello indicato; se si desidera trovare il file successivo che si accorda con quello indicato, bisogna chiamare la funzione **FindNext**.

Si osservi che queste due funzioni sono rispettivamente simili a Examine ed ExNext della libreria DOS, benché richiedano argomenti diversi. In particolare, FindFirst accetta come secondo argomento un puntatore a una struttura AnchorPath parzialmente inizializzata, e ne conclude automaticamente l'inizializzazione. FindNext accetta come suo unico argomento un puntatore alla struttura AnchorPath inizializzata da FindFirst.

La funzione **FreeAnchorChain** disalloca una struttura AnchorPath allocata da FindFirst o da FindNext. Torneremo su questo argomento in uno dei prossimi articoli.

La funzione **FileRequest** è *estremamente utile* in ogni programma che abbia a che fare con file e directory. FileRequest consente al programma di presentare un requester stile-Intuition, completo e a colori, che visualizza automaticamente tutti i file e le directory del file system dell'utente (eseguendo se necessario uno scroll) e permette all'utente di selezionare i file che intende utilizzare nel programma. Chiamando una sola funzione, si ottiene quindi una complessa serie di azioni software.

Per esempio, la funzione FileRequest è stata usata per creare il requester che appare quando si seleziona l'opzione Open nel programma *TxEd*. FileRequest mette a disposizione molte opzioni che permettono di confezionare un requester a misura del programma. Torneremo ancora su questo argomento in uno dei prossimi articoli.

La funzione **PathName** permette al programma di rintracciare il completo percorso di ricerca di un file o di una directory. PathName descrive completamente il percorso di ricerca, dalla radice del filing system fino al file.

La funzione **BaseName** fornisce un modo per ottenere la stringa del nome del file, partendo dalla più complessa stringa del percorso di ricerca. Per esempio, se un percorso di ricerca è: DF0:/xxx/yyy/zzz, il nome del file è zzz.

La funzione **TackOn** permette al programma di aggiungere il nome di un file a un percorso di ricerca. TackOn provvede anche ad aggiungere l'appropriato separatore (":" o "/").

14. La libreria ARP fornisce otto funzioni – ArpExit, LoadPrg, UnLoadPrg, CheckAbort, CheckBreak, SyncRun, ASyncRun e FindCLI – che possiamo classificare come funzioni di gestione della libreria ARP. Cioè queste funzioni possono essere usate per gestire i programmi ARP e le relazioni che intercorrono tra loro.

Si può utilizzare la funzione **ArpExit** per uscire da un programma in qualunque istante della sua esecuzione. Innanzitutto, ArpExit chiude la libreria ARP chiamando CloseLibrary, che a sua volta farà in modo di rilasciare tutte le risorse software sottoposte a track (liberandone la memoria). ArpExit segnala quindi al DOS di scaricare il programma dalla memoria, restituendo il codice d'errore di uscita al processo CLI e ogni indicazione sul motivo dell'uscita all'eventuale programma chiamante.

ArpExit restituisce un valore che può essere usato dal programma *Why* dell'AmigaDOS per permettere all'utente di saperne di più sul motivo dell'uscita dal programma. ArpExit svolge (con qualcosa in più) un'azione simile a quella della funzione "exit" dei sistemi *Manx* o *Lattice*.

La funzione **LoadPrg** carica un programma in memoria cercando un file predeterminato nel percorso di ricerca. Se viene chiamata da un programma mandato in esecuzione da CLI, LoadPrg per prima cosa cerca la lista residente ARP, poi il percorso specificato dall'utente tramite CLI. Se invece viene chiamata da un processo Workbench, l'ordine di ricerca è il seguente: primo, la lista residente ARP; secondo, la directory in cui si trova il programma in esecuzione; terzo, la directory logica C:.

La funzione **UnLoadPrg** scarica un programma caricato con LoadPrg.

Si noti che queste due funzioni sono rispettivamente simili a LoadSeg e UnLoadSeg della libreria DOS, con parecchie differenze che discuteremo in uno dei prossimi articoli.

La funzione **CheckAbort** può essere impiegata per controllare se l'utente ha digitato la combinazione Ctrl/C durante l'esecuzione del programma. Fornisce inoltre un meccanismo per chiamare

una funzione predefinita in linguaggio C, che può essere usata per visualizzare un messaggio sullo schermo prima dell'uscita dal programma.

Analogamente, la funzione CheckBreak permette al programma di controllare se l'utente ha inviato uno o più segnali di break, premendo una delle combinazioni di tasti: Ctrl/C/D/E/F. La funzione permette di specificare una maschera che lasci passare soltanto i segnali ai quali il programma deve reagire. Anche in questo caso, CheckBreak dà la possibilità d'indicare una funzione da chiamare nel momento in cui l'utente attiva uno dei segnali della maschera.

La funzione **SyncRun** permette a un programma di mandare in esecuzione un programma separato (processo o task) e aspetta il suo completamento prima di restituire il controllo (cioè manda in esecuzione il programma in modo sincrono a quello che ha chiamato la funzione).

SyncRun manda in esecuzione il programma da disco, ricercandolo nel percorso indicato dall'utente (secondo l'impostazione del comando Path dell'AmigaDOS). Quando il programma "figlio" è stato eseguito, SyncRun lo scarica dalla memoria e restituisce un codice di uscita per tornare al programma chiamante. Questa funzione può essere utilizzata soltanto dai programmi CLI (e non da quelli Workbench).

La funzione **ASyncRun** permette al programma di mandare in esecuzione un programma separato senza attendere necessariamente la sua conclusione per procedere con i codici del programma chiamante (esecuzione asincrona). La parola chiave, qui, è *senza attendere*. Per tale ragione, un simile processo "figlio" viene chiamato "processo in background".

Come SyncRun, ASyncRun manda in esecuzione il programma da disco, ricercandolo nel percorso indicato dall'utente secondo l'impostazione del comando Path dell'AmigaDOS. Si tratta di una funzione di una certa complessità, e anche su questo argomento torneremo nel prossimo articolo.

La funzione **FindCLI** fornisce al programma un puntatore a una struttura Process per uno specifico processo CLI. FindCLI ha come suo solo argomento il numero del processo (per esempio, il numero fornito dal comando STATUS dell'AmigaDOS) e restituisce un puntatore alla struttura Process che rappresenta il suddetto processo CLI.

15. La libreria ARP include cinque funzioni – AddResidentPrg, RemResidentPrg, CheckSumPrg, ObtainResidentPrg, ReleaseResidentPrg – per la gestione dei programmi residenti.

Ricordiamo che si dicono programmi residenti quei programmi che l'utente colloca nella lista residente del sistema ARP, usando, per esempio, il comando Resident del CLI. Queste funzioni permettono di gestire i programmi residenti senza uscire dal programma in esecuzione.

Il "gestore" dei programmi residenti ARP è una specie di software MMU (Memory Management Unit) che dà la possibilità di mandare in esecuzione più volte un programma, pur essendoci in memoria una sola copia dei suoi codici. Il risultato è una migliore efficienza, dovuta all'aumento della velocità d'esecuzione e alla minore richiesta di memoria.

La funzione **AddResidentPrg** aggiunge un dato programma alla lista di programmi residenti del sistema. Una volta installati, tutti i programmi (compreso quello che ha chiamato la funzione) possono mandare in esecuzione quello residente sfruttando la sua copia in memoria (vi sono alcune restrizioni sui programmi che si possono qualificare come residenti, ma torneremo sull'argomento nel prossimo articolo).

La funzione **RemResidentPrg** rimuove un programma dalla lista di sistema dei programmi residenti, rilasciando anche la memoria a esso associata. Se vi sono altri programmi che stanno ancora facendo uso del programma residente, la funzione non lo rimuove e restituisce invece un numero che indica quanti sono quei programmi.

La funzione **CheckSumPrg** fornisce un sistema per controllare che un programma appartenente alla lista dei programmi residenti ARP non sia stato variato in modo tale da non essere più ammissibile nella lista. Ricordiamo che le routine interne del sistema ARP calcolano automaticamente il checksum di tutti i programmi che vengono aggiunti alla lista dei residenti. CheckSumPrg mette a disposizione un meccanismo per ricalcolare il checksum in qualunque momento, e controllare che non sia cambiato. Nel caso che sia cambiato, il programma residente non è più accettabile e dovrebbe essere rimosso dalla lista.

La funzione **ObtainResidentPrg** permette al programma di accedere a un programma residente, cercandolo nell'apposita lista secondo il percorso di ricerca specificato. Se la funzione individua il programma cercato, incrementa il suo contatore d'uso (per indicare che c'è un altro programma che lo sta impiegando) e che per il momento non dev'essere rimosso (almeno finché il contatore non tornerà a zero).

La funzione **ReleaseResPrg** è l'inverso di ObtainResPrg. Fornisce un meccanismo per indicare al sistema ARP che il proprio programma non ha più bisogno di un particolare programma residente. ReleaseResPrg cerca nella lista residente la struttura Node del programma indicato e decrementa il suo contatore d'uso.

Nel prossimo articolo

Proprio come le funzioni in C o in Assembly, quelle della libreria ARP richiedono specifici argomenti e restituiscono specifici valori a seconda delle condizioni in cui si trova il sistema. Nel prossimo articolo illustreremo argomenti e valori restituiti, e spiegheremo in generale come queste funzioni vanno impiegate dai programmi. ■

UNA PARTITA A OTTO BIT

Inizia la sfida con il vostro C-64: scoprirete che anche un gioco di carte semplice come Dernière può diventare un passatempo divertente

di Robert Cook

D*uo*, la versione per C-64 del gioco di carte *Dernière* (chiamato anche *Uno*), è stata appositamente realizzata per due giocatori: voi e il computer. L'obiettivo è raggiungere o superare per primi la soglia dei cento punti. Si guadagnano punti quando si vince una mano di gioco, il che accade quando si esauriscono le carte a disposizione oppure quando l'avversario si ritrova con 18 o più carte in mano. È possibile giocare tramite tastiera o con un joystick collegato alla porta 2. Il mazzo del *Duo* contiene cento carte: otto gruppi di dieci carte (con valori da 0 a 9), sedici carte speciali (otto "Salta un Turno" e otto "Pesca due Carte") e quattro jolly. Ad eccezione di questi ultimi, le carte sono tutte suddivise in quattro colori: rosso, blu, verde e giallo. Il colore del jolly lo si può scegliere al momento di giocare la carta.

La prima mossa spetta al computer; in seguito è il vincitore della partita precedente a cominciare il gioco. All'inizio, il giocatore riceve sei carte e sei carte (coperte) vengono assegnate al computer; accanto al mazzo ne viene collocata una scoperta e su quella si effettueranno poi gli scarti.

Al proprio turno si può scartare, oppure prendere una carta dal mazzo. Nel primo caso si deve scartare una carta che abbia lo stesso colore, o valore o effetto (nel caso delle carte speciali "Salta un Turno" e "Pesca due Carte") di quella in cima alla pila degli scarti. Il jolly può essere invece giocato in qualsiasi momento (dichiarando il colore che gli si vuole attribuire).

Duo per C-64: la freccia indica la carta da giocare

Per scartare una carta, bisogna indicarla muovendo il cursore con la tastiera o con il joystick. Con la tastiera si usa il tasto cursore orizzontale per muoversi verso destra e il tasto cursore verticale per muoversi verso sinistra. Con il joystick bisogna spostare la levetta a destra o a sinistra. Per giocare la carta, si preme il tasto Return o il pulsante Fire. Se si tenta di giocare una carta non consentita, un segnale acustico indica al giocatore l'errore.

Se non si può o non si vuole scartare, si deve premere il tasto ↑, oppure muovere il joystick per spostare il cursore sul mazzo. Quindi si preme il tasto Return o il pulsante Fire per pescare una carta. Per spostare il cursore dal mazzo, si preme Shift/↑ o si muove il joystick verso il basso.

Dopo aver giocato una carta normale o un jolly (o aver pescato dal mazzo) viene il turno dell'avversario... a meno che non si giochi una carta "Salta un Turno" che fa "passare" l'avversario e fa restare il gioco in mano a noi. Se invece si gioca "Pesca due Carte" l'avversario deve pescare dal mazzo due carte extra.

Il jolly può essere giocato in qualsiasi momento. Dopo averlo giocato, si fanno scorrere i colori

e si indica il colore prescelto con il tasto Return o il pulsante Fire.

Quando uno dei giocatori termina le carte, il punteggio viene assegnato in base al valore delle carte rimaste in mano all'avversario. Se uno dei due giocatori si ritrova in mano diciotto o più carte, la partita finisce ugualmente e il punteggio si calcola facendo la differenza tra i punti che hanno in mano i due giocatori (purché chi raggiunge le diciotto carte abbia un punteggio superiore). Per le carte da 0 a 9 il valore è quello indicato dai numeri, le carte speciali valgono 20 punti, e i jolly 50.

A prescindere da come termina il gioco, al vincitore vanno aggiunti dieci punti di premio-extra, mentre al perdente non viene assegnato alcun punto. Una volta aggiornato il display del punteggio, si preme il tasto Return o il pulsante Fire per dare il via a un'altra partita. Se dal mazzo vengono pescate tutte le carte, la partita finisce senza vinti né vincitori.

Vince il giocatore che per primo raggiunge la quota di cento punti. Per giocare ancora si preme Y quando compare il prompt. Altrimenti, si preme N per tornare al sistema operativo.

Istruzioni: digitare il listato, salvarlo su disco e impartire il run.
Tutti i caratteri grafici e di controllo sono stati tradotti in combinazioni di tasti leggibili. Sono le istruzioni tra parentesi graffe.
Per ulteriori informazioni vedere "Come digitare i listati" a pagina 126.

```
10 REM DUO 64 - ROBERT B. COOK
20 POKE53280,11:POKE53281,1:POKE5326
5,11
30 GOSUB1620:GOSUB2040:POKE53265,27
40 GOSUB1010
50 POKE646,4-(PU=2)*4:PRINTEL$:ONPUG
OSUB90,510:POKESP,0
60 IFYC=0ORMC=0ORYC>17ORMC>17THENGO
SUB1210:GOTO40
70 IFDP=100THENGOSUB910:GOTO40
80 PU=3-PU:GOTO50
90 POKESX,30:POKESY,218:POKESP,1:MV=
1:SK=0:POKESB,0
100 PRINTLEFT$(CR$,7)"{3 CRSR →}{CTRL
9}{CTRL 5}{SHFT G}IOCA UNA CARTA O PES
CA DAL MAZZO{CTRL 0}":GOSUB950
110 GETG$:JS=PEEK(56320)AND31:IFG$="
"ANDJS=31THEN110
120 IFG$="↑"ORJS=30THENPOKESX,150:PO
KESY,170:POKESB,0:GOTO110
130 IFG$="{CRSR →}"ORJS=23THENMV=MV
+1:IFMV>YCTHENMV=1
140 IFG$="{CRSR ↓}"ORJS=27THENMV=MV-
1:IFMV<1THENMV=YC
150 FORL=1TO50:NEXT:T=14+MV*16:T1=0
:IFT>255THENT=T-255:T1=1
160 IFG$<>CHR$(13)ANDJS<>15THENPOKE
SX,T:POKESY,218:POKESB,T1:GOTO110
170 PRINTEL$
180 IFPEEK(SY)=170THENYC=YC+1:YC(YC
)=DK(DP):DP=DP+1:GOSUB320:RETURN
190 IFCV(YC(MV))<>12ANDCV(YC(MV))<>
CV(UC)ANDCC$(YC(MV))<>CC$(UC)THEN29
0
200 PRINTLEFT$(CR$,15)TAB(21)"{CTRL 2
}"+CD$(YC(MV))
210 PRINTLEFT$(CR$,15)TAB(21)CC$(YC(
MV))+CD$(YC(MV))
220 UC=YC(MV):YC=YC-1:IFYCTHENGOSUB
310:GOTO240
230 PRINTLEFT$(CR$,25)EC$;
240 IFCV(UC)=12ANDYCTHENGOSUB400
250 IFCV(UC)=11THENGOSUB460
260 IFCV(UC)=10ANDYCTHENGOSUB480
270 IFSK=1ANDYC>0ANDMC<18ANDDP<100T
HEN90
280 RETURN
290 PRINTLEFT$(CR$,7),"{CTRL 9}{CTRL 5
}{SHFT S}CEGLI UN'ALTRA CARTA{CTRL 0}":
GOSUB990
300 FORL=0TO999:NEXT:GOTO90
310 FORL=MVTOYC+1:YC(L)=YC(L+1):NEX
T:GOTO350
320 FORL=1TOYC-1:FORK=L+1TOYC
330 IFCV(YC(L))<CV(YC(K))THENT=YC(L)
:YC(L)=YC(K):YC(K)=T
340 NEXT:NEXT
350 T=0:FORL=1TOYC
360 IFL<19THENPRINTLEFT$(CR$,21)TAB(
(L-1)*2)CL$(T)+CC$(YC(L))+CD$(YC(L)
);
370 T=1:NEXT:IFYC<18THENPRINTEE$;
380 IFYC=19THENPRINTLEFT$(CR$,15)TAB
(34)CC$(YC(19))+CD$(YC(19));
390 RETURN
400 POKESP,0:T=0:PRINTLEFT$(CR$,7)TA
B(13)"{CTRL 9}{CTRL 5}{SHFT S}CEGLI UN C
OLORE{CTRL 0}":GOSUB950
410 GETG$:JS=PEEK(56320)AND31:IFG$="
"ANDJS=31THEN410
420 IF(G$=CHR$(13)ORJS=15)ANDT>0THE
NPRINT"{CTRL 5}"EL$:GOTO450
430 T=T+1+(T=4)*4:T$=MID$("{COMD 3}{
CTRL 6}{COMD 7}{CTRL 8}",T,1):WC$=T$:P
RINTLEFT$(CR$,15)TAB(21)T$CD$(UC)
440 CC$(UC)=T$:GOSUB950:GOTO410
450 WAIT56320,16:WAIT197,64:RETURN
460 PRINTLEFT$(CR$,7)TAB(12)"{CTRL 9}{
CTRL 5}{SHFT P}ESCO DUE CARTE{CTRL 0}"
:GOSUB950:FORS=0TO999:NEXT
470 FORJ=1TO2:IFDP<100THENMC=MC+1:M
C(MC)=DK(DP):DP=DP+1:GOSUB760:NEXT
480 IFYC=0ORMC>17ORDP>99THEN500
490 PRINTLEFT$(CR$,7),"{CTRL 9}{CTRL 5
}{SHFT T}OCCA ANCORA A TE{CTRL 0}":GOSU
B950:FORS=0TO999:NEXT
500 SK=-(DP<100):RETURN
510 PRINTLEFT$(CR$,7)TAB(14)"{CTRL 9}{
COMD 1}{SHFT E}' IL MIO TURNO{CTRL 0}":
GOSUB970:FORS=0TO999:NEXT
520 SK=0:MV=1:T=1
530 FORL=1TOMC:IFCV(MC(L))<10THEN580
540 IFCV(MC(L))<>CV(UC)ANDCC$(MC(L))
<>CC$(UC)ORCV(MC(L))=12THEN570
550 FORK=L+1TOMC:IFCC$(MC(K))=CC$(M
C(L))ORCV(MC(K))=CV(MC(L))THENMV=L:G
OTO640
560 NEXT
570 T=L+1:NEXT:IFYC<3THENT=1
580 FORL=TTOMC:IFCV(MC(L))=CV(UC)AND
CV(MC(L))<>12ANDMC(L)THENMV=L:GOTO6
40
590 IFCC$(MC(L))=CC$(UC)THENMV=L:GOT
O640
600 NEXT:IFT<>1THENT=1:GOTO580
610 IFCV(MC(1))=12THENMV=1:GOTO640
620 PRINTLEFT$(CR$,7)TAB(11)"{CTRL 9}{
COMD 1}{SHFT P}ESCO UNA CARTA    {CTRL
0}":GOSUB970:FORS=0TO999:NEXT
630 MC=MC+1:MC(MC)=DK(DP):DP=DP+1:G
OSUB760:RETURN
640 PRINTLEFT$(CR$,7)TAB(11)"{CTRL 9}{
COMD 1}{SHFT G}IOCO UNA CARTA{CTRL 0}":
GOSUB970:FORS=0TO999:NEXT
650 IFYC=1ANDCV(MC(1))=12THENMV=1
660 PRINTLEFT$(CR$,15)TAB(21)"{CTRL 2
}"+CD$(MC(MV));
670 PRINTLEFT$(CR$,15)TAB(21)CC$(MC(
MV))+CD$(MC(MV));
680 UC=MC(MV):MC=MC-1:IFMCTHENGOSUB
750:GOTO700
690 PRINTLEFT$(CR$,13)EC$
700 IFCV(UC)=12ANDMCTHENGOSUB810
710 IFCV(UC)=11THENGOSUB860
720 IFCV(UC)=10ANDYCTHENGOSUB880
730 IFSK=1ANDMC>0ANDYC<18ANDDP<100T
HEN520
740 RETURN
750 FORL=MVTOMC+1:MC(L)=MC(L+1):NEX
T:GOTO790
760 FORL=1TOMC-1:FORK=L+1TOMC
770 IFCV(MC(L))<CV(MC(K))THENT=MC(L)
:MC(L)=MC(K):MC(K)=T
780 NEXT:NEXT
790 IFMC<19THENPRINTLEFT$(CR$,9)TAB(
(MC-1)*2)BK$;:IFMC<18THENPRINTEE$;
800 RETURN
810 T1=0:T$="":FORL=1TO4:T=0:FORK=1
TOMC
820 IFCC$(MC(K))=MID$("{COMD 3}{CTRL 6
}{COMD 7}{CTRL 8}",L,1)THENT=T+1
830 NEXT:IFT>T1THENT1=T:T$=MID$("{CO
MD 3}{CTRL 6}{COMD 7}{CTRL 8}",L,1)
840 NEXT:IFT$=""THENT$=MID$("{COMD 3
```

```
}{CTRL 6}{COMD 7}{CTRL 8}",INT(RND(1)*4+
1),1)
850 PRINTLEFT$(CR$,15)TAB(21)T$+CD$(
UC):CC$(UC)=T$:RETURN
860 PRINTLEFT$(CR$,7)TAB(11)"{CTRL 9}{
COMD 1}{SHFT P}ESCA DUE CARTE{CTRL 0}":
GOSUB970:FORS=0TO999:NEXT
870 FORJ=1TO2:IFDP<100THENYC=YC+1:Y
C(YC)=DK(DP):DP=DP+1:GOSUB320:NEXT
880 IFMC=0ORYC>17ORDP>99THEN900
890 PRINTLEFT$(CR$,7)TAB(11)"{CTRL 9}{
COMD 1}{SHFT T}OCCA ANCORA A ME{CTRL 0}
":GOSUB970:FORS=0TO999:NEXT
900 SK=-(DP<100):RETURN
910 PRINTLEFT$(CR$,19)TAB(14)EC$:PRI
NTLEFT$(CR$,7)"{CTRL 4}"EL$
920 PRINTLEFT$(CR$,7)TAB(7)"{CTRL 9}{S
HFT C}ARTE FINITE, PARTITA TERMINATA"
930 FORL=1TO6:GOSUB970:GOSUB950:NEXT
940 GOSUB1440:RETURN
950 POKESR,240:POKEWV,17:POKEHF,50:P
OKELF,35:POKEAT,0
960 FORS=1TO99:NEXT:POKEWV,16:POKESR
,0:RETURN
970 POKEVL,15:POKEAT,0:POKESR,225:PO
KEWV,17:POKEHF,33:POKELF,33
980 FORS=1TO99:NEXT:POKEWV,16:RETURN
990 POKEHF,5:POKEAT,0:POKESR,240:POK
EWV,33
1000 FORS=1TO500:NEXT:POKEWV,32:RETU
RN
1010 T=RND(-TI):SYS900
1020 PRINTLEFT$(CR$,7)"{CTRL 4}"EL$:PR
INTLEFT$(CR$,7)TAB(11)"{CTRL 9}{SHFT M
}ESCOLO LE CARTE{CTRL 0}"
1030 GOSUB950:GOSUB970
1040 FORL=1TO100:T=INT(RND(1)*100+1
):T1=DK(L):DK(L)=DK(T):DK(T)=T1:NEX
T
1050 FORL=1TO19:YC(L)=0:MC(L)=0:NEX
T
1060 T=0:FORL=1TO12STEP2:T=T+1:YC(T
)=DK(L):MC(T)=DK(L+1):NEXT
1070 FORL=1TO5:FORK=L+1TO6
1080 IFCV(YC(L))<CV(YC(K))THENT=YC(L
):YC(L)=YC(K):YC(K)=T
1090 IFCV(MC(L))<CV(MC(K))THENT=MC(L
):MC(L)=MC(K):MC(K)=T
1100 NEXT:NEXT:PRINTLEFT$(CR$,7)"{CTR
L 4}"EL$
1110 T=0:FORL=1TO6
1120 PRINTLEFT$(CR$,21)TAB((L-1)*2)C
L$(T)+CC$(YC(L))+CD$(YC(L));
1130 PRINTLEFT$(CR$,9)TAB((L-1)*2)BK
$
1140 T=1:NEXT{16 SPAZI}
1150 IFCV(DK(13))<10THEN1170
1160 T1=INT(RND(1)*85+16):T=DK(13):
DK(13)=DK(T1):DK(T1)=T:GOTO1150
1170 PRINTLEFT$(CR$,15)TAB(14)BK$
1180 PRINTLEFT$(CR$,15)TAB(21)CC$(DK
(13))+CD$(DK(13))
1190 MC=6:YC=6:UC=DK(13):DP=14
1200 RETURN
1210 IFMC=0THEN1250
1220 T=0:FORL=1TOMC
1230 IFL<19THENPRINTLEFT$(CR$,9)TAB(
(L-1)*2)CL$(T)+CC$(MC(L))+CD$(MC(L)
)
1240 T=1:NEXT
1250 POKE646,4-(MC=0ORYC>17)*4:PRIN
TLEFT$(CR$,7)TAB(2)EL$
1260 T=0:FORL=0TOYC:T1=CV(YC(L))
1270 IFT1<10THENT=T+T1
1280 IFT1=10ORT1=11THENT=T+20
1290 IFT1=12THENT=T+50
1300 NEXT:YT=T
1310 T=0:FORL=0TOMC:T1=CV(MC(L))
1320 IFT1<10THENT=T+T1
1330 IFT1=10ORT1=11THENT=T+20
1340 IFT1=12THENT=T+50
1350 NEXT:MT=T
1360 IFYC>17THENYT=YT-MT:IFMT<0THEN
MT=0
1370 IFMC>17THENMT=MT-YT:IFYT<0THEN
YT=0
1380 IFYC>0ANDMC<18THEN1420
1390 PRINTLEFT$(CR$,7)TAB(10)"{CTRL 9
}{SHFT I}L TUO PUNTEGGIO:"MT+10"PUNTI
":GOSUB950:GOSUB950
1400 YS=YS+MT+10:PU=2:IFYS>99THENYG
=YG+1
1410 IFMC>0ANDYC<18THEN1440
1420 PRINTLEFT$(CR$,7)TAB(11)"{CTRL 9
}{SHFT I}L MIO PUNTEGGIO:"YT+10"PUNTI
":GOSUB970:GOSUB970
1430 MS=MS+YT+10:PU=1:IFMS>99THENMG
=MG+1
1440 FORL=0TO2999:NEXT:FORL=97TO100:
CC$(L)="{CTRL 1}":NEXT:SK=0:GOSUB1590
1450 POKE198,0:IFMS>99ORYS>99THEN150
0
1460 PRINTLEFT$(CR$,7)"{CTRL 4}"EL$:PR
INTLEFT$(CR$,7)TAB(10)"{CTRL 9}{SHFT P
}REMI {SHFT F}{SHFT I}{SHFT R}{SHFT E} O
{SHFT R}{SHFT E}{SHFT T}{SHFT U}{SHFT R}{S
HFT N}"
1470 GOSUB950:GOSUB970
1480 GETG$:JS=PEEK(56320)AND31:IFG$<
>CHR$(13)ANDJS<>15THEN1480
1490 RETURN
1500 IFYS>MSTHENPRINTLEFT$(CR$,7)TAB
(4)"{CTRL 9}{CTRL 5}{SHFT H} A I' V I N
T O  T U!{9 SPAZI}"
1510 IFMS>YSTHENPRINTLEFT$(CR$,7)TAB
(6)"{CTRL 9}{COMD 1}{SHFT H} O  V I N T
O  I O!{11 SPAZI}"
1520 POKE896,MG:POKE897,YG:POKE898,P
U:GOSUB1580
1530 FORL=1TO10:GOSUB950:GOSUB970:NE
XT:FORL=0TO3999:NEXT
1540 PRINTLEFT$(CR$,7)"{CTRL 4}"EL$:PR
INTLEFT$(CR$,7)TAB(11)"{CTRL 9}{SHFT G
}IOCHI ANCORA? [Y/N]"
1550 GOSUB950:GOSUB970
1560 POKE198,0:WAIT198,1:GETG$:IFG$<
>"N"THENRUN
1570 END:GAME
1580 MG=PEEK(896):YG=PEEK(897):PU=P
EEK(898)
1590 PRINTLEFT$(CR$,3)TAB(27)"{CTRL 9
}{COMD 4}"RIGHT$(" "+STR$(MS),3)TAB(34
)MG
1600 PRINTLEFT$(CR$,4)TAB(67)"{CTRL 9
}{COMD 4}"RIGHT$(" "+STR$(YS),3)TAB(74
)YG
1610 RETURN
1620 DIM T,T1,T2,L,K,J,DP,MC,YC,MT,Y
T,MS,YS,MG,YG,PU,MV,JS,SK,G$,T$,WC$
1630 DIM CD$(100),CC$(100),CV(100),D
K(100),MC(20),YC(20)
1640 BK$="{CTRL 4}{COMD A}{3 SHFT *}{COM
D S}{5 CRSR ←}{CRSR ↓}{SHFT -}{CTRL 9}{S
HFT D}{SHFT £}{COMD *}{CTRL 0}{SHFT -}{5 C
RSR ←}{CRSR ↓}{SHFT -}{CTRL 9}{SHFT £}{SH
FT U}{SHFT £}{CTRL 0}{SHFT -}{5 CRSR ←}{C
RSR ↓}{SHFT -}{CTRL 9}{COMD *}{SHFT £}{SHF
T O}{CTRL 0}{SHFT -}{5 CRSR ←}{CRSR ↓}{C
OMD Z}{3 SHFT *}{COMD X}"
1650 EE$="{CTRL 9}{COMD 4}{2 SPAZI}{2 CR
SR ←}{CRSR ↑}{2 SPAZI}{2 CRSR ←}{CRSR ↑
}{2 SPAZI}{2 CRSR ←}{CRSR ↑}{2 SPAZI}{CTR
L 0}"
1660 EC$="{CTRL 9}{COMD 4}{5 SPAZI}{5 CR
SR ←}{CRSR ↑}{5 SPAZI}{5 CRSR ←}{CRSR ↑
}{5 SPAZI}{5 CRSR ←}{CRSR ↑}{5 SPAZI}{5 C
RSR←}{CRSR ↑}{5 SPAZI}{CTRL 0}"
1670 EL$="{HOME}{6 CRSR ↓}{2 CRSR →}{CT
RL 9}{36 SPAZI}{CTRL 0}"
1680 CL$(1)="{COMD 5}{CRSR ←}{COMD N}{C
RSR ←}{CRSR ↓}{COMD N}{CRSR ←}{CRSR ↓}{C
OMD N}{CRSR ←}{CRSR ↓}{COMD N}{CRSR ←}{C
RSR ↓}{COMD N}{4 CRSR ↑}"
1690 E$="{5 CRSR ←}{CRSR ↓}":CR$="{HO
ME}{24 CRSR ↓}"
1700 LF=54272:HF=54273:WV=54276:AT=
54277:SR=54278
1710 SP=53269:SX=53248:SY=53249:SB=
53264
1720 POKE54296,15:POKE2040,13:POKE53
287,0:POKESP,0
1730 POKE53276,1:POKE53285,3:POKE532
77,0:POKE53271,1
1740 FORL=1TO96STEP8:C$="":FORK=1TO
4:READD$:C$=C$+D$+E$:NEXT:READD$
1750 FORK=0TO7:CD$(L+K)=C$+D$:CC$(L
+K)=MID$("{2 COMD 3}{2 CTRL 6}{2 COMD
7}{2 CTRL 8}",K+1,1):CV(L+K)=T
1760 NEXT:T=T+1:NEXT
1770 C$="":FORK=1TO4:READD$:C$=C$+D
$+E$:NEXT:READD$
1780 FORL=97TO100:CD$(L)=C$+D$:CC$(
L)="{CTRL 1}":CV(L)=12:NEXT
1790 FORL=1TO100:DK(L)=L:NEXT
1800 IFPEEK(1024)=160THEN1840
1810 FORL=832TO897:READK:POKEL,K:NEX
T
1820 FORL=900TO933:READK:POKEL,K:NEX
T
1830 PU=1:POKE898,1
1840 RETURN
1850 DATA "0{3 SPAZI}0"," {CTRL 9}{COMD
 V} {COMD C}{CTRL 0} "," {CTRL 9} {CTRL 0
} {CTRL 9} {CTRL 0} "," {CTRL 9}{COMD F}
{COMD D}{CTRL 0} ","0{3 SPAZI}0"
1860 DATA "1{3 SPAZI}1"," {COMD D}{CTRL
 9} {CTRL 0}{2 SPAZI}","{2 SPAZI}{CTRL 9}
 {CTRL 0}{2 SPAZI}"," {COMD D}{CTRL 9} {C
TRL 0}{COMD F} ","1{3 SPAZI}1"
1870 DATA "2{3 SPAZI}2"," {CTRL 9}{COMD
 V}{COMD O}{COMD C}{CTRL 0} "," {CTRL 9}{2
```

```
COMD U}{COMD D}{CTRL 0} "," {CTRL 9} {2
COMD U}{CTRL 0} ","2{3 SPAZI}2"
1880 DATA "3{3 SPAZI}3"," {CTRL 9}{COMD
 V}{COMD O}{COMD C}{CTRL 0} ","{2 SPAZI}{S
HFT *}{CTRL 9}{SHFT *}{CTRL 0} "," {CTRL
9}{COMD F}{COMD U}{COMD D}{CTRL 0} ","3{3
SPAZI}3"
1890 DATA "4{3 SPAZI}4"," {CTRL 9}{COMD
 Y}{COMD K}{CTRL 0}{COMD K} "," {CTRL 9}
{COMD V}{COMD C}{CTRL 0} ","{2 SPAZI}{CTR
L 9}{COMD K}{CTRL 0}{COMD K} ","4{3 SPAZI
}4"
1900 DATA "5{3 SPAZI}5"," {CTRL 9} {2 C
OMD P}{CTRL 0} "," {CTRL 9}{2 COMD P}{COM
D C}{CTRL 0} "," {CTRL 9}{COMD F}{COMD Y}
{COMD D}{CTRL 0} "," 5{3 SPAZI}5"
1910 DATA "6{3 SPAZI}6"," {CTRL 9}{COMD
 V}{COMD O}{COMD C}{CTRL 0} "," {CTRL 9}
{CTRL 0}{SHFT *}{CTRL 9}{COMD U}{CTRL 0} "
," {CTRL 9}{COMD F}{COMD Y}{COMD D}{CTRL
0} ","6{3 SPAZI}6"
1920 DATA "7{3 SPAZI}7"," {CTRL 9}{COMD
 @}{COMD O} {CTRL 0} ","{2 SPAZI}{CTRL 9}
{COMD K}{CTRL 0}{COMD K} ","{2 SPAZI}{CTR
L 9} {CTRL 0}{2 SPAZI}","7{3 SPAZI}7"
1930 DATA "8{3 SPAZI}8"," {CTRL 9}{COMD
 V}{COMD O}{COMD C}{CTRL 0} "," {CTRL 9}{S
HFT *}{CTRL 0}{SHFT *}{CTRL 9}{SHFT *}{CTR
L 0} "," {CTRL 9}{COMD F}{COMD U}{COMD D}
{CTRL 0} ","8{3 SPAZI}8"
1940 DATA "9{3 SPAZI}9"," {CTRL 9}{COMD
 V}{COMD O}{COMD C}{CTRL 0} "," {CTRL 9}{C
OMD O}{CTRL 0}{SHFT *}{CTRL 9} {CTRL 0} "
," {CTRL 9}{COMD F}{COMD U}{COMD D}{CTRL
0} ","9{3 SPAZI}9"
1950 DATA "{SHFT S}{4 SPAZI}","K{CTRL 9
}{COMD V}{COMD P}{COMD @}{CTRL 0}{SHFT T}"
,"I{CTRL 9}{COMD F}{SHFT A}{COMD C}{CTRL
0}U","P{CTRL 9}{COMD T}{COMD Y}{COMD D}{CT
RL 0}R","{4 SPAZI}N"
1960 DATA "{SHFT D}{3 SPAZI}{SHFT C}","R
{CTRL 9} {COMD I}{COMD C}{CTRL 0}A","A{CT
RL 9} {CTRL 0} {CTRL 9} {CTRL 0}R","W{CTR
L 9} {CTRL 0}{COMD I}{CTRL 9}{COMD D}{CTRL
 0}D","2{3 SPAZI}S"
1970 DATA "{SHFT W}{4 SPAZI}","I{CTRL 9
} {CTRL 0} {CTRL 9} {CTRL 0}{SHFT C}","L{C
TRL 9}{COMD G}{CTRL 0}{SHFT -}{CTRL 9}{COM
D M}{CTRL 0}A","D{CTRL 9}{COMD J}{SHFT -
}{COMD L}{CTRL 0}R","{4 SPAZI}D"
1980 DATA 170,170,128,149,085,128,14
9,106,128,149,106,128,149,090,000,14
9
1990 DATA 086,128,153,085,160,154,08
5,104,154,149,090,170,165,090,168,16
9
2000 DATA 106,000,042,168,000,010,16
0,000,002,128,000,000,000,000,000,00
0
2010 DATA 000,000,000,000,000,000,00
0,000,000,000,000,000,000,000,0,
0,0
2020 DATA 162,170,169,11,157,63,217,
157,233,217,157,147,218,157,61,219,1
69
2030 DATA 160,157,63,5,157,233,5,157
,147,6,157,61,7,202,208,225,96
2040 PRINT"{SHFT CLR}{CTRL N}{CTRL 9}{CO
MD 4}{18 SPAZI}{COMD A}{20 SHFT *}{COMD S
}";
2050 PRINT"{CTRL 0}{COMD F}{COMD D}{2 CO
MD I}{COMD C}{CTRL 9} {CRSR →}{3 SPAZI}{C
RSR →} {CTRL 0}{COMD V}{3 COMD I}{COMD C
}{CTRL 9} {SHFT -}{8 SPAZI}{SHFT P}UNTEGG
IO{2 SPAZI}{SHFT -}";
2060 PRINT"{CTRL 0}{COMD K}{CTRL 9}{COMD
 K}{2 SPAZI}{CRSR →} {CRSR →}{3 SPAZI}{C
RSR →} {CRSR →}{3 SPAZI}{CRSR →}{SHFT
 -} {SHFT C}OMP:{13 SPAZI}{SHFT -}";
2070 PRINT"{CTRL 0}{COMD V}{COMD C}{CTRL
 9}{2 COMD I}{CTRL 0}{COMD D}{CTRL 9} {CTR
L 0}{COMD F}{CTRL 9}{3 COMD I}{CTRL 0}{COM
D D}{CTRL 9} {CTRL 0}{COMD F}{CTRL 9}{3 CO
MD I}{CTRL 0}{COMD D}{CTRL 9} {SHFT -} {SH
FT T}U:{15 SPAZI}{SHFT -}";
2080 PRINT"{CTRL 9}{18 SPAZI}{COMD Z}{20
 SHFT *}{COMD X}";
2090 PRINT"{CTRL 9}{2 SPAZI}{36 COMD @}
{2 SPAZI}";
2100 PRINT"{CTRL 9} {COMD M} {36 SPAZI}{
COMD G}";
2110 PRINT"{CTRL 9}{2 SPAZI}{36 COMD T}{
2 SPAZI}"
2120 GOSUB1580:SYS900:RETURN
```

■

UN 128 PIÙ VELOCE IN OGNI MODALITÀ

Come accelerare i programmi nel modo 40 colonne del C-128 anche se si è in modo 64

di Bret Timmins

Una delle più apprezzabili caratteristiche del C-128 rispetto ai precedenti computer Commodore è la maggior velocità. D'altra parte, nel modo a 40 colonne la velocità extra ha un prezzo: il VIC II, il chip grafico che gestisce il display a 40 colonne non funziona correttamente a velocità superiori a 1 MHz (modo Slow), e dev'essere disattivato quando il computer lavora a 2 MHz (modo Fast).

Per aggirare questa situazione, ho scritto *Fast-40*, un sintetico programma in linguaggio macchina per il C-128. *Fast-40* accelera da un minimo del venti fino a un massimo del cento per cento la velocità dei programmi che funzionano nel modo 40 colonne, sia quelli in Basic sia quelli in linguaggio macchina. Per ottenere questo risultato, cancella parte dello schermo (in una quantità che dipende dall'aumento di velocità richiesto).

Ci sono due diverse versioni di *Fast-40*: per il modo 128 (listato 1) e per il modo 64 (listato 2).

Sempre più in fretta

Benché niente sembri cambiato, dopo aver mandato in esecuzione *Fast-40* il vostro C-128 sta elaborando i dati a una velocità del 20 per cento più alta rispetto alla normale velocità del modo a 40 colonne: questo è il massimo aumento di velocità ottenibile con *Fast-40* se volete continuare a vedere l'intero schermo.

Per aumentare ulteriormente la velocità d'elaborazione, premete insieme i tasti Control e +. Una riga grigio chiaro coprirà la prima riga dello schermo. Premendo ancora Ctrl/+ viene cancellata la seconda riga dello schermo, e così via: ogni riga cancellata rappresenta un incremento di velocità del 3-4 per cento. Ctrl/+ può continuare a essere premuto finché non resta libera soltanto un riga di schermo, che corrisponde a un incremento di velocità del 98 per cento.

Per diminuire la velocità e vedere una parte più estesa dello schermo, premete Ctrl/–.

Nella stesura di un programma si può impostare la velocità a cui far operare *Fast-40* impartendo un comando POKE 6415 nella versione del modo 128, o POKE 52515 nella versione 64, con un valore compreso tra 50 e 242. *Fast-40* usa questo valore per determinare quante righe di schermo deve cancellare. Il valore di default è 50 (nessuna riga cancellata e incremento di velocità del 20 cento). Il valore massimo è 242 (24 righe cancellate e incremento di velocità del 98 per cento). Per avere aggiornamenti di schermo senza intoppi, i valori da inserire nella locazione di memoria devono essere numeri tra 50 e 242 incrementati di 8 a partire dal 50: 58, 66, 74 e così via.

Volendo accelerare ancora, si può ottenere una velocità di 2 MHz premendo Ctrl/F: l'intero schermo viene cancellato e assume il colore del bordo. Per tornare al modo 1 MHz basta premere Shift/Ctrl/F. Quando si è nel modo Program si può attivare il modo Fast completo con il comando SYS 6403 per la versione 128, oppure SYS 52503 per la versione 64; SYS 6406 (128) o SYS 52506 (64) per disattivarlo.

Precauzioni

Benché io abbia fatto tutto il possibile per evitare possibilità di crash, ci sono occasioni in cui è bene disabilitare *Fast-40*. Le periferiche (disk drive, stampanti, modem...) richiedono sincronizzazioni precise per operare correttamente e *Fast-40* può alterare quei tempi. Nel modo 128 non è un gran problema, perché il sistema operativo imposta automaticamente il modo a 1 MHz ogni volta che accede a una periferica. Comunque, lo schermo diventa confuso.

Nel modo 64, il sistema operativo pensa di lavorare sempre a 1 MHz e se durante l'accesso a una periferica *Fast-40* è attivo, il sistema si blocca. *Fast-40* cerca di prevenire questa situazione disabilitandosi ogni volta che viene impartito un comando Load, Save o Open (sia in Basic che in linguaggio macchina), ma alcuni programmi aggirano questi comandi e chiamano le periferiche tramite altre routine. Con programmi di questo tipo, *Fast-40* va in crash, a meno che non venga disabilitato prima di ogni operazione con le periferiche.

Il comando Ctrl/X serve a disattivare temporaneamente *Fast-40* mentre il 128 comunica con le periferiche (per riattivarlo si deve usare Shift/Ctrl/X). Nel modo Program si usa SYS 6409 (SYS 52509 nel modo 64) per disabilitare *Fast-40*, e si usa invece il suo indirizzo di partenza (SYS 6400 per il modo 128 e SYS 52500 per il modo 64) per riattivarlo.

Dal momento che è un programma basato sugli interrupt, occasionalmente può non riuscire a effettuare un cambiamento da un modo grafico all'altro (dal modo grafico ad alta risoluzione al modo testo, per esempio). È un'eventualità rara, e succede soltanto nel modo 64. Anche in questo caso si deve usare Ctrl/X per disabilitare temporaneamente *Fast-40* prima di ogni cambiamento grafico.

Si osservi infine che, nel modo 128, i modi Split-Screen Graphics 2 e 4 non sono disponibili con *Fast-40*.

Un programma basato sugli interrupt

Fast-40 funziona ordinando al C-128 di generare due interrupt di raster (uno in fondo allo schermo e uno nei pressi della parte alta). L'interrupt del fondo imposta il modo 2 MHz, controlla l'eventuale pressione di tasti Control e dice al VIC di attivare la grafica ad alta risoluzione.

Per qualche incomprensibile motivo, la grafica ad alta risoluzione visualizzata dal VIC a 2 MHz appare sullo schermo sotto forma di righe

color grigio-chiaro. L'interrupt che si verifica nella parte alta della bitmap ristabilisce il modo a 1 MHz ed effettua il reset del chip VIC. Si osservi che il listato pubblicato qui di seguito *non è* il programma definitivo, si tratta del programma che "crea" il vero *Fast-40* in linguaggio macchina. Di conseguenza, dopo aver digitato il listato e dopo averlo salvato su disco, impartire il run ha soltanto l'effetto di creare il file eseguibile.

Per caricare il file in memoria e mandarlo in esecuzione si utilizzano i comandi:

```
BLOAD "FAST-40.128"
SYS 6400
```

per quanto riguarda il modo 128. In modo 64 si deve digitare:

```
LOAD "FAST-40.64",8,1:
NEW:SYS 52500.
```

Istruzioni: digitare il listato, salvarlo su disco e impartire il run.
Tutti i caratteri grafici e di controllo sono stati tradotti in combinazioni di tasti leggibili. Sono le istruzioni tra parentesi graffe.
Per ulteriori informazioni vedere "Come digitare i listati" a pagina 126.

Fast-40 modo C-128

```
0 REM QUESTO LISTATO CREA IL FILE DA
 UTILIZZARE: FAST40-128
5 OPEN 8,8,8,"FAST40-128,P,W"
6 CT=0:PRINT"{SHFT CLR}"
10 READ A$:IF A$="-1" THEN CLOSE8:PR
INT:PRINT"TUTTO FATTO!":END
12 PRINT"{HOME}LEGGO RIGA "+STR$(CT):
CT=CT+1
15 IF LEN(A$)<62 THEN 55
20 B$=MID$(A$,1,20)+MID$(A$,22,20)+
MID$(A$,43,20)
25 FOR I=1 TO 30
30 C$=MID$(B$,(I*2)-1,2):H$=LEFT$(C
$,1):L$=RIGHT$(C$,1)
35 H=VAL(H$):IF H$>"9" THEN H=ASC(H
$)-55
40 L=VAL(L$):IF L$>"9" THEN L=ASC(L
$)-55
45 BY=H*16+L:PRINT#8,CHR$(BY);
50 NEXT:GOTO 10
55 IF LEN(A$)<21 THEN B$=A$:GOTO 70
60 IF LEN(A$)<42 THEN B$=LEFT$(A$,20
)+RIGHT$(A$,(LEN(A$)-21)):GOTO 70
65 B$=LEFT$(A$,20)+MID$(A$,22,20)+R
IGHT$(A$,LEN(A$)-42)
70 FOR I=1 TO LEN(B$)/2
75 C$=MID$(B$,(I*2)-1,2):H$=LEFT$(C
$,1):L$=RIGHT$(C$,1)
80 H=VAL(H$):IF H$>"9" THEN H=ASC(H
$)-55
85 L=VAL(L$):IF L$>"9" THEN L=ASC(L
$)-55
90 BY=H*16+L:PRINT#8,CHR$(BY);
95 NEXT:GOTO 10
100 REM FAST-40 MODO 128 LM
101 DATA 00194C1219A9C02CA901*2CA980
8DDB1A6032FA00*78A9478D1403A9198D15
102 DATA 03AD10198D12D0AD11D0*297F8D
11D0AD1AD00901*8D1AD0A9FF8D19D0A902
103 DATA 8D11198DDA1AA9008DDB*1A5860
D8AD19D029018D*19D0AC1119B9D21A8D61
104 DATA 19C8B9D21A8D62194C00*00AD16
D029EF0DD71A8D*16D0AD11D029DF0DD61A
105 DATA 8D11D0A9008D30D0A902*8D1119
AD10198D12D0AD*11D0297F8D11D04C33FF
106 DATA A9018D30D0AD11D00920*8D11D0
AD16D029EF8D16*D020DC1AA0052CDB1A70
107 DATA 6B3050A5D3C904D04DA5*D4C940
F047C928D01ACE*D81AD03E8CD81AAD0F19
108 DATA C9F2F03418AD0F196908*8D0F19
D029C92BD01ACE*D91AD0208CD91AAD0F19
109 DATA C932F01638AD0F19E908*8D0F19
D00BC915F01DC9*17D0034C7A1AA9008D11
110 DATA 19AD0F198D12D0AD11D0*297F8D
11D04CC71AA94C*8D1403A91A8D1503AD11
111 DATA D029EF8D11D0AD11D029*DF0DD6
1A8D11D0AD16D0*29EF0DD71A8D16D0202C
112 DATA 1B4CC71AAD19D029018D*19D0AD
DB1AC901F00CA5*D3C905D0E5A5D4C915D0
113 DATA DFA9008D30D0AD11D009*108D11
D02012194C33FF*A9008D30D08DDA1AAD16
114 DATA D029EF0DD71A8D16D0AD*11D029
DF0DD61A09108D*11D0A9AA8D1403A91A8D
115 DATA 1503202C1B4CC71AAD19*D02901
8D19D0A5D3C905*D00CA5D4C917D0062012
116 DATA 194CC71A202C1BA9008D*DB1A20
2CC24C69FA6319*9419000005050000A5D8
117 DATA 4829208DD61AF010AD2D*0A8D18
D0A50129FD0904*8501D010AD2C0A8D18D0
118 DATA A50129FB090205D98501*682980
F002A9108DD71A*ADDA1AD016AD11D029DF
119 DATA 0DD61A8D11D0AD16D029*EF0DD7
1A8D16D06020DC*1AAD10198D12D0AD11D0
120 DATA 297F8D11D060
121 DATA -1
```

Fast-40 modo C-64

```
0 REM QUESTO LISTATO CREA IL FILE DA
 UTILIZZARE: FAST40-64
5 OPEN 8,8,8,"FAST40-64,P,W"
6 CT=0:PRINT"{SHFT CLR}"
10 READ A$:IF A$="-1" THEN CLOSE8:PR
INT:PRINT"TUTTO FATTO!":END
12 PRINT"{HOME}LEGGO RIGA "+STR$(CT):
CT=CT+1
15 IF LEN(A$)<62 THEN 55
20 B$=MID$(A$,1,20)+MID$(A$,22,20)+
MID$(A$,43,20)
25 FOR I=1 TO 30
30 C$=MID$(B$,(I*2)-1,2):H$=LEFT$(C
$,1):L$=RIGHT$(C$,1)
35 H=VAL(H$):IF H$>"9" THEN H=ASC(H
$)-55
40 L=VAL(L$):IF L$>"9" THEN L=ASC(L
$)-55
45 BY=H*16+L:PRINT#8,CHR$(BY);
50 NEXT:GOTO 10
55 IF LEN(A$)<21 THEN B$=A$:GOTO 70
60 IF LEN(A$)<42 THEN B$=LEFT$(A$,20
)+RIGHT$(A$,(LEN(A$)-21)):GOTO 70
65 B$=LEFT$(A$,20)+MID$(A$,22,20)+R
IGHT$(A$,LEN(A$)-42)
70 FOR I=1 TO LEN(B$)/2
75 C$=MID$(B$,(I*2)-1,2):H$=LEFT$(C
$,1):L$=RIGHT$(C$,1)
80 H=VAL(H$):IF H$>"9" THEN H=ASC(H
$)-55
85 L=VAL(L$):IF L$>"9" THEN L=ASC(L
$)-55
90 BY=H*16+L:PRINT#8,CHR$(BY);
95 NEXT:GOTO 10
100 REM FAST-40 MODO 64 LM
101 DATA 14CD4C26CDA9C02CA901*2CA980
8D44CF6032FA00*A9C78D1803A9CE8D1903
102 DATA A9E68D1A03A9CE8D1B03*A9D88D
3003A9CE8D3103*A9E08D3203A9CE8D3303
103 DATA A97F8D0DDCA9838D1403*A9CD8D
1503AD24CD8D12*D0AD11D0297F8D11D0AD
104 DATA 1AD009018D1AD0A9FF8D*19D0A9
028D25CDA9008D*44CF60AD19D08D19D0AC
105 DATA 25CDB93CCF8D9ACDC8B9*3CCF8D
9BCD4C0000A900*8D30D0AD16D029EF0D41
106 DATA CF8D16D0AD11D029DF0D*40CF8D
11D0A9028D25CD*AD24CD8D12D0AD11D029
107 DATA 7F8D11D04C81EAA9018D*30D0AD
11D029208D40CF*AD16D029108D41CFAD11
108 DATA D009208D11D0AD16D029*EF8D16
D0A0052C44CF70*6C3051AD8D02C904D04D
109 DATA A5C5C940F047C928D01A*CE42CF
D03E8C42CFAD23*CDC9F2F03418AD23CD69
110 DATA 088D23CDD029C92BD01A*CE43CF
D0208C43CFAD23*CDC932F01638AD23CDE9
111 DATA 088D23CDD00BC915F01D*C917D0
0320F4CEA9008D*25CDAD23CD8D12D0AD11
112 DATA D0297F8D11D04C31EAA9*9D8D14
03A9CE8D1503A9*008D1AD08D44CFA9FF8D
113 DATA 0DDCAD11D029EF8D11D0*AD11D0
29DF0D40CF8D11*D0AD16D029EF0D41CF8D
114 DATA 16D04C31EAAD44CFC901*F00DAD
8D02C905D0EFA5*C5C915D0E9A9008D30D0
115 DATA 8D44CFAD11D009108D11*D02026
CD4C81EA488A48*9848A591C97FD00320F4
116 DATA CE4C4CFE859320F4CE4C*A7F420
F4CE4CEDF5A5BA*F007C903F00320F4CE4C
117 DATA 4AF3A9008D30D08D44CF*AD16D0
29EF0D41CF8D16*D0AD11D029DF0D40CF09
118 DATA 108D11D0A9298D1403A9*CF8D15
03A9008D1AD0A9*FF8D0DDC60AD8D02C905
119 DATA D009A5C5C917D0032026*CD4C31
EA9CCDCDCD0000*050500
120 DATA -1
```

CORSO DI PROGRAMMAZIONE IN C PER L'AMIGA: SETTIMA PUNTATA

LE PROCEDURE D'AZIONE NEI PROGRAMMI INTUITION

Il modo migliore per imparare è studiare i listati dei programmi. "Sezioniamo" quindi il programma Testo per fare i primi passi nella costruzione di un serio programma stile-Intuition in C

di Eugene P. Mortimore

Nell'articolo precedente abbiamo presentato e descritto un programma "stile Intuition" che apriva due finestre (una finestra grafica e una finestra testo), visualizzava un disegno nella finestra grafica, la chiudeva automaticamente quando il disegno era completo e infine consentiva semplici operazioni di menu nella finestra testo.

Ora lasceremo temporaneamente il reame della grafica e concentreremo l'attenzione esclusivamente sul testo.

Per iniziare è necessario affrontare il concetto di "procedure d'azione": si tratta di un gruppo di operazioni correlate (più esattamente un gruppo di funzioni o routine in linguaggio C) che vanno in esecuzione quando l'utente seleziona la voce di un menu tramite il pulsante destro del mouse.

Nel nostro programma, l'evento "selezione della voce di un menu" veniva elaborato dalle routine interne di Intuition, che a questo scopo inviavano il numero della voce di menu alla funzione ProcessMenuItems. Ricevuto il numero della voce, la funzione iniziava una serie di azioni software per raggiungere l'obiettivo voluto dall'utente. Il concetto di "procedure d'azione" è fondamentale per i programmi Intuition, ed è uno dei principi basilari nella progettazione dei programmi in genere.

Mentre si procede nel lavoro di programmazione, ci si rende conto ben presto che i codici sorgente di qualunque programma Intuition (o, se è per questo, di qualunque programma di qualche interesse, a parte le piccole utility) si dilatano progressivamente fino a diventare molto ingombranti. Questa considerazione, comunque, vale specialmente per i programmi Intuition, che definiscono un'interfaccia utente grafica (schermi, finestre, requester, menu, gadget...).

Il problema, nel nostro caso, è quello di gestire un testo di grandi dimensioni, problema che diventa particolarmente complesso quando c'è più di un file da gestire. Per risolverlo nel modo migliore bisogna seguire una procedura quanto più sistematica possibile.

Si inizia la costruzione

Benché il nostro obiettivo immediato sia l'ampliamento di un programma Intuition, vorremmo anche soffermarci sulle generiche procedure da seguire per costruire un programma Intuition in modo sistematico. A questo scopo sottolineiamo subito il seguente concetto: è bene rimandare all'ultimissima fase della programmazione tutti gli orpelli grafici. Nelle fasi iniziali o intermedie conviene progettare schermi, finestre, menu, voci di menu, requester e gadget semplici e funzionali, che incorporino tutte le caratteristiche essenziali richieste dal programma. Ogni abbellimento conviene rimandarlo all'ultimo momento.

Se non si procede secondo questa linea di principio, si finisce per rimanere impantanati nei dettagli delle immagini grafiche e inevitabilmente la logica del programma ne soffre.

Prima di mettersi a scrivere le funzioni in linguaggio C che faranno parte delle procedure d'azione del programma (ovvero le funzioni che vanno in esecuzione quando l'utente seleziona la voce di un menu) è importante aver deciso fino in fondo tutte le operazioni che dovranno compiere i menu. In particolare, aver deciso i dettagli dei loop di elaborazione dei messaggi IDCMP della classe MENUPICK (relativi cioè alle operazioni dei menu).

A questo scopo conviene partire sempre con "funzioni di prova" durante la progettazione delle operazioni dei menu, rimandando a un secondo momento l'idea di aggiungere alle funzioni tutti i dettagli secondari.

Le funzioni di prova sono funzioni designate per avere argomenti "di prova" (le cui caratteristiche possono cambiare in ogni momento, per fornire al programma un comportamento quanto più possibile funzionale) che restituiscono semplicemente il controllo al punto del programma in cui sono state chiamate. Concentrate i vostri sforzi iniziali nell'identificare gli argomenti appropriati, e lavorate con le funzioni di prova quanto più a lungo è possibile, verificando tutte le possibili sequenze e combinazioni di input dell'utente.

Solo quando il menu principale e tutti i loop di elaborazione degli eventi sono stati sottoposti a un completo debug, è il caso di dedicarsi alle singole funzioni, aggiungendole una dopo l'altra al programma, ricompilandolo e ricontrollando tutte le operazioni ogni volta.

All'inizio conviene collocare tutte le funzioni di prova in fondo al listato, e metterle al loro posto soltanto quando arriva il momento di ampliarle nella forma finale. Questa tecnica permette di semplificare la compilazione, e parallelamente permette ai menu (e a tutte le altre caratteristiche di base del programma) di funzionare. In tal modo il programmatore può assicurarsi che gli aspetti principali della logica del programma siano a posto, prima di affrontare il livello successivo: le singole funzioni.

Quando una particolare funzione svolge correttamente il proprio compito, conviene escluderla dallo svolgimento del programma (utilizzando i delimitatori che servono in genere per i commenti) e continuare a utilizzare quella di prova. Si passa quindi alla funzione successiva. Consigliamo di mettere in disparte la funzione completa principalmente per evitare di trovarsi a ricompilare più volte funzioni già programmate correttamente.

Poi, alla fine, quando tutte le funzioni sono state pienamente definite e verificate, si sostituiscono le funzioni di prova con le funzioni definitive e si controllano le operazioni del programma nel loro complesso. Gli ulteriori ritocchi, a questo punto, dovrebbero essere una cosa veloce, dal momento che la maggior parte delle basi logiche sono già sicuramente corrette.

Altri suggerimenti sulla programmazione

Quando state scrivendo funzioni per una procedura d'azione (o per qualunque altra sezione di programma) sentitevi sempre liberi di fare esperimenti, e se qualcosa non funziona la prima volta tentate qualcos'altro, ma senza spostarvi troppo da quella riga del programma sorgente.

Per operare nel modo più utile, escludete dal programma le linee che rappresentano i primi tentativi (o anche i seguenti) così che siano recuperabili nel caso che in seguito si rivelino utili.

A questo riguardo, è importante notare che i compilatori della Lattice e della Manx, tramite apposite opzioni, permettono entrambi di nidificare i commenti, cosicché anche se si aggiunge una nuova coppia di "/* */" all'interno di una coppia già esistente, si può essere sicuri che la compilazione verrà eseguita correttamente.

Allo stesso modo, se c'è una sezione di codice che probabilmente non verrà utilizzata nel programma definitivo (ma che può essere utile magari in un altro programma già previsto), conviene racchiuderla in un blocco di testo e sfruttare poi la funzione MOVE TEXT dell'editor per spostarlo in fondo al listato, e infine delimitarlo con i soliti "/* */". In tal modo, come tutte le sezioni di codice racchiuse dai delimitatori di commento, quel blocco non rallenterà la compilazione.

Questa procedura, che potremmo chiamare di "commento" assicura che durante la ricompilazione il compilatore non perda tempo con quelle linee, e contemporaneamente assicura che quelle linee vengano salvate in un file su disco, a disposizione del programmatore nel caso che risultassero utili in un altro momento.

Se usate l'editor *TxEd*, ecco un breve sommario di alcuni dei comandi più utili al programmatore.

La combinazione Amiga-destro/K "elimina" la linea su cui si trova il cursore, compreso il carattere Return che la conclude. Per far ricomparire la riga nel punto in cui si trova il cursore si usa invece la combinazione Amiga-destro/L.

Quindi, se dovete copiare una riga da un punto all'altro, basta collocare il cursore sulla riga (in qualunque punto), digitare Amiga-destro/K e poi Amiga-destro/L per farla ricomparire, e poi muovere il cursore nella nuova locazione e digitare di nuovo Amiga-destro/L.

Se si vuole trasferire un testo in un file sorgente aperto, "importandolo" da un altro file, si deve: 1) utilizzare il comando MORE TXED per aprire una seconda finestra testo sullo schermo; 2) rendere attiva la finestra ; 3) digitare Amiga-destro/O per chiamare il File Requester; 4) selezionare il nome del secondo file; 5) caricarlo nella seconda finestra; 6) esaminare la seconda finestra finché non si trova il blocco di testo da importare; 7) evidenziare il blocco con il comando Amiga-destro/B; 8) copiarlo nel Clipboard con il comando Amiga-destro/C; 9) rendere attiva la prima finestra; 10) muovere il cursore nella prima finestra dove si vuole trasferire il blocco, e digitare Amiga-destro/I.

Ci si può muovere avanti e indietro tra le due finestre e utilizzare ripetutamente questa procedura, arrivando addirittura a costruire un nuovo file sorgente solo assemblando varie parti di file già esistenti.

Un'altra idea interessante è quella d'impostare il tabulatore alle colonne 2, 4, 6, 8... (cioè ogni due

colonne) per produrre le rientranze del codice (dette anche indentazioni) in punti fissi e garantire una miglior leggibilità del programma. Se si usano spazi di tabulazione più grandi, per esempio ogni otto colonne, le righe di codice indentate più profondamente rischiano di trovarsi troppo a destra nella pagina.

La scelta che abbiamo proposto noi, invece, permette di far stare la maggior parte delle righe nelle 80 colonne dello schermo o della stampante. Anche così, però, alcune righe saranno più lunghe di 80 caratteri; suggeriamo quindi di utilizzare la stampa compressa per fare in modo che, nei limiti del possibile, le righe non superino mai la larghezza del foglio.

Potrebbe anche far comodo sostituire i complicati nomi delle funzioni dell'Amiga con nomi più esplicativi come ProcessMenuItems; i codici sono senz'altro più leggibili e più comprensibili se si divide in blocchi il codice sorgente (come nel caso del listato che proponiamo al termine dell'articolo). Sempre a questo scopo – la leggibilità – sono state inserite alcune righe di commento con gli opportuni delimitatori ai lati.

I programmi di testo

Se si vuole lavorare con il testo in una finestra di Intuition, conviene imparare ad aprire e a utilizzare il dispositivo Console. Le sue routine interne, infatti, portano alla finestra testo parecchie utility che permettono di visualizzare e gestire un testo senza doversi creare un sistema di routine personali. Le routine del dispositivo Console sono un esempio lampante dell'utilità delle routine predefinite dell'Amiga.

Facciamo un esempio: senza le routine del dispositivo Console, per effettuare il "wordwrap" delle parole vicino ai bordi della finestra (bordi che possono essere spostati dall'utente in qualsiasi momento) si sarebbe costretti a scrivere una routine per tenere continuamente sott'occhio il margine destro della finestra testo.

In altre parole, se il dispositivo Console non viene attivato, ogni volta che l'utente sposta o ridimensiona la finestra testo le routine del programma devono esplicitamente spostare all'inizio della riga successiva ogni carattere che oltrepassa il margine destro. Per quanto non sia affatto impossibile, una simile soluzione richiede un ampio lavoro di programmazione che non porterebbe vantaggi tali da compensare lo sforzo.

Per di più sarebbe necessario scrivere un complicato loop di elaborazione per messaggi IDCMP, al fine di elaborare i dati grezzi dei caratteri di tastiera inviati dalle routine interne del dispositivo Input.

Perché questo loop funzioni correttamente, si deve tenere conto di tutte le diverse classi di caratteri (quelli stampabili e quelli di controllo, come Line Feed, Return e così via) ed elaborare individualmente ognuno di essi tramite complessi loop di messaggi IDCMP.

Anche questo è fattibile, naturalmente, e inoltre consentirebbe l'introduzione di caratteristiche non disponibili facendo uso delle routine del dispositivo Console, ma alla fine del lavoro ci si renderebbe conto che per la maggior parte dei programmi i vantaggi non valgono lo sforzo. Per il programmatore medio è senza dubbio consigliabile l'uso delle caratteristiche previste dalle routine del dispositivo Console per gestire i testi nelle finestre.

Una volta che si sia deciso di fare uso delle routine del dispositivo Console, il problema successivo è trasferire nella finestra i testi salvati su disco, e visualizzarli. Vi sono diversi modi per affrontarlo.

Per prima cosa, si possono utilizzare le funzioni di I/O in linguaggio C relative a caratteri e stringhe (fopen, fclose, getc, gets, scanf...) che si trovano in tutte le librerie C standard. Queste routine leggono i caratteri immagazzinati in appositi buffer di testo e li inviano alle routine del dispositivo Console per la visualizzazione.

In modo simile (ma restando più vicini allo spirito dell'Amiga) si può eseguire la stessa operazione ricorrendo alle routine per la gestione dei file messe a disposizione dall'AmigaDOS. Sono routine inserite direttamente in ROM, come Open, Read, Write, Close, che servono a leggere file di dati in buffer di testo su disco e a trasferirli nella finestra tramite le routine del dispositivo Console. Questa è la tecnica che abbiamo utilizzato nel programma d'esempio.

Si inizia, come vedrete, chiamando la funzione Open dell'AmigaDOS, poi trasferendo i caratteri in buffer di memoria tramite la funzione Read, scrivendo i caratteri nel file destinazione tramite la funzione Write e infine chiudendo il file con la funzione Close dell'AmigaDOS.

Il listato d'esempio

Il listato *Testo* mostra un programma esemplificativo che apre una finestra testo, la collega al dispositivo Console e vi trasferisce alcune voci di un database, prendendoli da un file "MioDatabase" creato in precedenza. In effetti, quest'ultimo file potrebbe essere stato creato proprio utilizzando il database illustrato nell'articolo su *F-Basic*, pubblicato in questo stesso numero della rivista.

Proponiamo il listato *Testo* come punto di partenza per creare un programma più esteso, che consenta all'utente di definire e aggiornare le voci di un database.

Rispetto al listato *Menu.c*, pubblicato nel numero 1/90, sono stati fatti i seguenti ampliamenti e modifiche.

- Tutti i riferimenti a MyWindow2 (la finestra grafica) sono stati eliminati; di conseguenza abbiamo potuto eliminare ogni riferimento ai file INCLUDE della libreria Graphics.
- La grandezza iniziale di MyWindow1 è stata

aumentata fino a occupare quasi l'intero schermo al di sotto della barra dei menu. Si noti il cambiamento nella definizione della struttura NewWindow (nel parametro Height che è stato aumentato a 160).

• Ulteriori voci di menu (dichiarazioni di strutture MenuItem) sono state aggiunte al menu della finestra MyWindow1; in particolare sono state aggiunte sei voci all'opzione User, allo scopo di fornire all'utente un sistema per gestire i record del database. Come si noterà, le opzioni proposte sono molto simili a quelle che compaiono nello schermo iniziale del programma *Rubrica* (in appendice all'articolo su *F-Basic*). Nel listato, le strutture MenuItem che rappresentano le nuove voci sono quelle che vanno da MenuItem10 a MenuItem5. Queste strutture MenuItem sono concatenate "all'indietro", come quelle del menu Project.

• È stata definita una nuova struttura DBList, una struttura in linguaggio C la cui definizione coincide esattamente con quella della struttura List usata nel programma *Rubrica*. In effetti avremmo potuto utilizzare lo stesso nome di struttura (List) ma la presenza della definizione di struttura List nei file INCLUDE dell'Exec avrebbe causato un conflitto di nomi.

Invece i nomi dei parametri utilizzati all'interno della struttura DBList sono gli stessi utilizzati nella struttura List (data, name, phone e link). Sempre seguendo la falsariga del programma *Rubrica*, abbiamo definito due puntatori ("head" e "nextp") per puntare a specifici esemplari di strutture DBList. L'altro puntatore utilizzato nel programma *Rubrica*, "trail", in questa versione preliminare del programma non serve.

• Abbiamo inoltre aggiunto tre "#define" (MsgPort, Request e Device) alla lista che rappresenta l'indice delle risorse software utilizzate dal programma. Si noti che i numeri assegnati, a differenza dei numeri delle voci di menu di cui parliamo nel prossimo paragrafo, proseguono *verso l'alto* rispetto alle definizioni contenute nel listato precedente.

Tali definizioni assegnano a ogni risorsa software un numero-indice che corrisponde a un bit impostato a 1 in una specifica locazione. Per esempio, la definizione DEVICE (64) corrisponde a un bit a 1 al settimo posto. Queste definizioni servono per impostare a 1 questi specifici bit quando viene aperta con successo la relativa risorsa di programma: nel momento in cui una risorsa è necessaria, si va a controllare lo stato di quel bit.

• Sono stati aggiunti cinque nuovi #define relativi alle voci dei menu (da READ a ERASE) per fissare i numeri che la funzione ProcessMenuItems userà per elaborare i messaggi IDCMP provenienti dalle routine di Intuition, ovvero quando l'utente esegue una selezione dal menu User. Si noti che questi valori partono da 0 e quindi non sono una continuazione dei numeri progressivi fissati per le voci del menu Project. Ne approfittiamo per mettere in chiaro che ciascun menu ha un proprio insieme di numeri progressivi per le voci dei menu. Se avessimo definito i nuovi numeri partendo da 5 anziché da 0, il programma non avrebbe funzionato: provare per credere. Torneremo su questi indici quando discuteremo le modifiche alla funzione ProcessMenuItems.

• Nel segmento "main()" del programma si continua ad avere a che fare con due sole variabili locali: "class" e "code". Anche in questo caso sono necessarie variabili locali per rappresentare i parametri Class e Code degli esemplari di strutture IntuiMessage, parametri che vengono definiti automaticamente dalle routine interne di Intuition, ogni volta che l'utente fa una selezione da un menu.

• Nel segmento "main()" sono stati fatti anche altri cambiamenti. Innanzitutto è stata usata la funzione CreatePort per creare una message port tramite la quale le routine interne del dispositivo Console possono dialogare con il programma. Si noti che il puntatore MyMsgPort è una variabile locale che punta a quello specifico esemplare di struttura MsgPort.

Allo stesso modo, si è impiegata la funzione CreateStdIO per creare uno specifico esemplare di struttura IOStdReq che viene usata più avanti nel corso del programma. Il puntatore a quella struttura IOStdReq è MyRequest.

L'inizializzazione della struttura IOStdReq

Una volta ottenuta la struttura IOStdReq, inizializziamo i parametri io_Data e io_Length: io_Data viene inizializzato per puntare alla struttura MyWindow1 (il cui indirizzo è il valore restituito dalla chiamata alla funzione OpenWindow), mentre io_Length viene inizializzato con la dimensione della struttura MyWindow1.

L'inizializzazione viene svolta dalle due seguenti istruzioni:

```
MyRequest->io_Data = (APTR) MyWindow1;
MyRequest->io_Length = sizeof(*MyWindow1);
```

Ricordiamo che questa coppia di istruzioni è una convenzione di programmazione dell'Amiga per collegare una finestra di Intuition al dispositivo Console.

L'apertura di una risorsa

Eseguito il collegamento, dobbiamo aprire il dispositivo. Questo è lo scopo dell'istruzione successiva:

```
OpenDevice("console.device", 0L, MyRequest, 0L);
```

Si noti che la chiamata alla funzione OpenDevice della libreria Exec utilizza come terzo argomento il puntatore alla struttura MyRequest preceden-

temente fornito dalla chiamata alla funzione CreateStdIO.

La precedente sequenza di tre istruzioni dovrebbe essere usata ogni volta che si deve collegare una finestra già aperta di Intuition (tramite OpenWindow) con il dispositivo Console.

Si deve poi controllare che la risorsa software aperta sia effettivamente disponibile. Questi "test di disponibilità" sono molto simili alla procedura utilizzata per controllare se le chiamate alle funzioni OpenLibrary, OpenScreen e OpenWindow hanno avuto successo. In questo programma abbiamo, in totale, sei tentativi di aprire risorse software e ogni volta si deve controllare se l'operazione ha avuto successo; in caso contrario il programma non proseguirebbe. Una delle cause d'insuccesso più frequenti è la carenza di memoria per l'allocazione di una struttura; un altro motivo è la non disponibilità delle opportune librerie (come la libreria Intuition o quella del dispositivo Console).

Si noti che quando una risorsa software è stata aperta con successo, viene impostato immediatamente un bit (per esempio il bit DEVICE) nel parametro "mask"; in seguito, durante lo svolgimento della funzione CloseAll, questi bit vengono controllati per sapere quali risorse devono effettivamente essere chiuse.

Si tratta di una procedura semplice e rapida per tenere il conto di tutte le varie risorse software impiegate dal programma. Ogni risorsa in più richiede soltanto che il programmatore aggiunga la corrispondente definizione (#define) per un bit del parametro mask, e la corrispondente istruzione Open nel punto appropriato. Se si desidera aprire il dispositivo Input, per esempio, si definisce un nuovo bit di mask (#define INPUT 128) e si aggiunge un'altra chiamata alla funzione OpenDevice subito dopo quella per aprire il dispositivo Console. Poi si controlla il risultato della chiamata (procedendo come nel caso del controllo per il dispositivo Console) e in caso di risultato positivo si imposta il bit INPUT. La chiusura della risorsa nei codici della funzione CloseAll è analoga a quella del dispositivo Console.

Vediamo che cosa succede se il programma non trova la risorsa richiesta. Supponiamo per esempio che la variabile MyScreen risulti NULL; il programma allora visualizza il messaggio "Impossibile aprire lo schermo", chiama la funzione CloseAll e torna all'interfaccia linea comando. In altre parole, se la chiamata alla funzione Open non ha successo (perché, per esempio, non c'è abbastanza memoria per allocare la struttura Screen), nel puntatore MyScreen viene memorizzato il valore NULL e non viene impostato alcun bit del parametro mask. Il programma visualizza quindi l'appropriato messaggio e chiama CloseAll. D'altra parte, poiché la mancata apertura si è verificata alla prima chiamata, nessuno dei bit delle risorse è impostato e quindi CloseAll restituisce semplicemente il controllo, senza chiamare né CloseScreen, né CloseWindow né alcun'altra funzione di chiusura: il programma torna quindi al CLI poco dopo aver iniziato l'esecuzione.

Come viene usato il dispositivo Console

Come già abbiamo detto, il dispositivo Console adatta automaticamente il testo alla finestra di Intuition, rispettando i margini destro e sinistro. Se l'utente ridimensiona la finestra, sono le routine del dispositivo Console che si occupano di effettuare l'appropriato wordwrap del testo. Il wordwrap e lo scroll verticale automatico sono i due maggiori vantaggi che si ottengono usando il dispositivo Console per visualizzare un testo in una finestra Intuition.

Nel nostro programma abbiamo creato la funzione ConsolePrint, perché provvedesse a visualizzare le stringhe a terminazione nulla *appropriatamente* definite. Ogni stringa di testo, infatti, dev'essere appropriatamente preparata prima di essere affidata alla funzione ConsolePrint. Come si osserva dal listato, la definizione di ConsolePrint è situata quasi alla fine; la funzione accetta due argomenti: il primo è un puntatore a una struttura IOStdReq e il secondo è un puntatore a una stringa a terminazione nulla. Non appena ConsolePrint ha ricevuto questi due valori, provvede a inizializzare ulteriormente la struttura IOStdReq nel modo seguente: nel parametro io_Data viene memorizzato l'indirizzo della stringa a terminazione nulla, nel parametro io_Length viene memorizzato il valore 1L, e nel parametro io_Command viene memorizzato il valore CMD_WRITE. Ricordiamo che il valore numerico del parametro CMD_WRITE è definito nei file INCLUDE console.device; quindi basta indicare il suo valore simbolico.

Inizializzati questi valori, ConsolePrint usa la funzione DoIO() della libreria Exec per inviare il comando CMD_WRITE alle routine interne del dispositivo Console. Quindi in realtà è la funzione DoIO che provvede a visualizzare il testo sulla finestra Intuition. Anche se può sembrare un sistema un po' contorto per visualizzare un testo in una finestra di Intuition, si tratta ancora una volta di una conseguenza dell'esistenza delle routine interne dell'Amiga (in questo caso quelle del dispositivo Console) e delle regole predefinite per accedervi.

Si noti che quando una delle sei chiamate "Open" fallisce, il testo viene visualizzato nella finestra CLI di partenza (ovvero quella da cui è stato mandato in esecuzione il programma Intuition).

Vorremmo sottolineare che non è possibile visualizzare messaggi di errore tramite il dispositivo Console, per segnalare nella finestra testo la mancata apertura di una risorsa. Infatti il dispositivo Console non diventa disponibile finché l'ultima risorsa di programma (ovvero lo stesso dispositivo

Console) non viene aperta dalla chiamata a OpenDevice. Il programma usa quindi sei istruzioni di chiamata alla funzione "printf" della libreria C standard, per stampare gli appropriati messaggi d'errore nella finestra CLI. A questo punto si deve provvedere a liberare memoria (o prendere qualunque altro provvedimento si riveli necessario per correggere la condizione d'errore) e rimandare in esecuzione il programma.

Questa discussione è importante perché mette in luce che non si possono utilizzare le risorse delle routine interne del dispositivo Console finché il dispositivo stesso non è stato aperto e collegato alla finestra. Non è sufficiente la presenza della funzione ConsolePrint. Se si tenta di visualizzare il testo nella finestra prima del momento opportuno, il programma non può funzionare. È con questo concetto ben chiaro in mente che sono stati scritti i codici relativi ai bit della variabile mask, quelli relativi all'apertura delle risorse software, e quelli della funzione CloseAll.

Ora che la finestra è correttamente impostata, si può iniziare a visualizzare il testo impiegando ConsolePrint. Il primo "write", subito dopo la chiamata a OpenDevice, serve semplicemente a inizializzare la finestra per le successive operazioni di scrittura; si noti che a questo scopo il programma scrive nella finestra testo una stringa a terminazione nulla, definita dall'array di caratteri "ConsolePrep".

Questa stringa a terminazione nulla ha una definizione molto particolare: consiste di otto byte seguiti da un byte NULL (0x00). Andando più a fondo, si osserva che la stringa contiene due sottosequenze: la prima inizia con il carattere 0x9B, a cui fanno seguito i caratteri 0x35, 0x35 e 0x79; la seconda inizia ugualmente con il carattere 0x9B, e seguono i caratteri 0x30, 0x20 e 0x70. Dopo la "stampa" di questa stringa (che risulta invisibile sulla finestra), il programma stampa un messaggio per avvisare l'utente che ogni ulteriore carattere visualizzato nella finestra vi viene situato dalla funzione ConsolePrint tramite le routine interne del dispositivo Console. Si tratta semplicemente di un promemoria per informare l'utente su un dettaglio che riguarda il funzionamento del programma.

Le variazioni nell'elaborazione dei menu

Nel precedente articolo abbiamo presentato un completo loop di elaborazione delle voci di menu per il menu Project, e abbiamo illustrato come le routine del programma e quelle di Intuition interagissero per elaborare le selezioni fatte dall'utente tramite il mouse.

Ora illustreremo il loop aggiunto per l'elaborazione delle sei ulteriori voci che formano il menu User. Dunque, sotto l'intestazione "User" abbiamo sei voci, da READ a ERASE. Ricordiamo che i loro valori sono stati stabiliti nei dati globali di programma, prima del segmento "main()" e vanno da 0 a 5: ogni volta che l'utente fa una selezione dal menu User, le routine interne di Intuition forniscono un valore intero alla variabile "item" e l'appropriata sezione di codici va in esecuzione. Sia chiaro che i nomi associati alle voci di menu tramite l'istruzione #define sono semplici etichette per i numeri effettivamente restituiti al programma dalle routine interne di Intuition. Servono soltanto a rendere il programma più facile da leggere e da interpretare.

Le funzioni di prova

Come abbiamo sottolineato in precedenza, nella stesura dell'interfaccia a menu di un programma applicativo conviene utilizzare funzioni di prova, ovvero funzioni con argomenti posticci, che vengono chiamate e restituiscono il controllo senza svolgere nessuna operazione utile. Una volta che l'interfaccia ha raggiunto uno stadio finale, le funzioni di prova vengono sostituite da quelle vere.

Il nostro esempio dimostra sistematicamente come procedere. Si noti che la funzione ProcessMenuItems incorpora chiamate a sei funzioni di elaborazione relative ai record del database: da ReadRec a EraseRec. Nel nostro caso, per mantenere il programma entro dimensioni ragionevoli abbiamo riportato nel listato soltanto la versione estesa della funzione ReadRec, le altre cinque funzioni rimangono presenti solo come funzioni di prova.

Lo switch doppiamente nidificato qui utilizzato è istruttivo da studiare come esempio di complessa istruzione SWITCH in linguaggio C. Si osservi che entrambi i "casi" principali (Project e User) si concludono con un'istruzione "default", e che tra i due casi c'è un'istruzione "break". Non si dimentichi mai d'includere quel piccolo "break" tra i due casi principali: la sua mancanza può causare uno strano comportamento da parte del programma. Si provi a eliminarla, e si osservi come il programma elabora i messaggi IDCMP relativi ai menu e quali messaggi compaiono nella finestra Intuition quando si selezionano le voci dei menu. Riuscite a spiegarvi un simile risultato?

La struttura di ProcessMenuItems può essere facilmente estesa per includere tutte le voci appartenenti agli altri cinque menu del programma. Basta aggiungere un "caso" principale (per esempio un caso Edit) per ciascun menu, e alcuni casi minori per le voci appartenenti ai casi principali.

Non dimenticate l'istruzione "default" alla fine di ciascun caso principale e l'istruzione "break" tra l'uno e l'altro. Seguendo questa falsariga si può costruire un programma piuttosto complesso, a metà tra un database e un elaboratore di testi (operando con le funzioni di prova finché non si ottiene dal programma il comportamento desiderato).

Il nostro semplice programma Testo

Come abbiamo già detto, nel listato viene presentata nel suo aspetto definitivo soltanto la funzione ReadRec. ReadRec ha lo scopo di leggere da un file di nome MioDatabase un insieme di record (il file MioDatabase può essere stato creato con un qualunque word processor, o anche con il programma *Rubrica* presentato al termine dell'articolo su *F-Basic*). Per essere coerenti con il programma *Rubrica*, supponiamo che ogni record del file abbia il seguente aspetto:

k,nnnnnnnnnn,pppppppppp(Return)

Cioè, ciascun record ha una cifra indice (k) seguita da una virgola, poi ci sono i dieci caratteri del campo "nome", una virgola, i dieci caratteri del campo "numero telefonico" e infine un carattere Return. I campi "nome" e "numero telefonico" possono eventualmente contenere spazi vuoti. Tutti i caratteri, inclusi quelli nulli, vengono trattati come caratteri di testo: per leggere il record, il programma usa la funzione Read della libreria DOS, che tratta appunto tutti i caratteri di un record come caratteri di testo.

Vediamo ora qual è il modo di operare della funzione ReadRec.

Per prima cosa, l'utente seleziona l'opzione OPEN del menu Project. Dal listato del programma si osserva subito che l'effetto è una chiamata alla funzione OpenFile, la quale a sua volta chiama la funzione Open della libreria DOS per aprire il file su disco MioDatabase. Da questo momento in poi (finché MioDatabase non verrà chiuso) tutti i successivi input di record arriveranno da questo file, ogni volta che l'utente seleziona l'opzione READ del menu User.

Si noti che OpenFile imposta il parametro puntatore "head" a NULL per inizializzare la lista di record del database che verrà costruito "dinamicamente" (aggiungeremo in seguito questo particolare) trasferendo i record dal disco alla memoria. Anche questa procedura è molto simile a quelle che abbiamo utilizzato nel già citato programma *Rubrica*.

Si noti che OpenFile usa la funzione Open della libreria DOS per aprire un file esistente ("mode" = MODE_OLDFILE) e fornisce al programma un "filehandle" (fh) come riferimento al file. Dai dati che precedono l'inizio del programma main(), osserviamo che la variabile "fh" è definita come un puntatore a una struttura FileHandle, una variabile globale. La struttura FileHandle è definita nel file INCLUDE dos.h.

Ora, se il programma non può aprire il file MioDatabase, lo indica all'utente stampando un messaggio nella finestra testo. Si tratta di un messaggio d'errore stampato nella finestra Intuition tramite la funzione ConsolePrint (anziché la funzione printf). In realtà, il messaggio d'errore può essere visualizzato in questa finestra perché il dispositivo Console è già stato aperto, e la relativa finestra è stata opportunamente preparata per la visualizzazione.

Questa coppia di funzioni – OpenFile e ReadRec – mostra come la libreria DOS, residente in ROM, venga utilizzata per portare un testo in una finestra Intuition e formattarlo tramite le routine del dispositivo Console.

Via via che si trasferiscono nella finestra i successivi record, si può osservare in azione lo scroll verticale automatico. Eventualmente si provi a restringere la finestra per vedere come il testo viene automaticamente mandato a capo dalla funzione di wordwrap.

Le funzioni ReadRec e Read

Supponiamo che il file MioDatabase sia stato aperto senza problemi: ora l'utente può trasferirvi (leggere) un insieme di record predefiniti selezionando la voce READ del menu User. Possiamo osservare che la funzione ReadRec opera nel modo seguente.

Innanzitutto definisce un insieme di sei variabili locali: cinque variabili testo ("k", "c1", "c2", "newname" e "newphone") e una variabile numerica che contiene solo numeri interi ("ccount", contatore di caratteri). Quindi effettua una serie di chiamate alla funzione Read della libreria DOS per leggere i record del file MioDatabase (per una spiegazione completa della funzione Read, si veda *Il Manuale dell'AmigaDOS*, pubblicato dalla IHT Gruppo Editoriale).

La funzione Read richiede tre argomenti: prima di tutto il filehandle fornito dalla chiamata alla funzione Open della libreria DOS. Si noti che la variabile "fh" è definita come globale (cioè è valida per l'intero programma), ed è automaticamente disponibile per la funzione ReadRec dal momento in cui viene eseguita OpenFile.

Il secondo argomento è l'indirizzo del buffer di testo dove i caratteri dovranno essere provvisoriamente trasferiti dalla funzione Read, in preparazione al trasferimento al dispositivo Console. Il terzo argomento è il numero di caratteri che dovranno essere trasferiti.

La variabile "ccount" (il cui valore viene restituito dalla chiamata a Read) serve per verificare che la funzione abbia trasferito effettivamente il numero richiesto di caratteri.

Più comunemente, come accade anche nel nostro caso, il programma controlla soltanto che il valore di ccount non sia –1 (si tratta, ancora una volta, di una convenzione dell'AmigaDOS). Se Read restituisce –1 nella variabile ccount, il programma visualizza un messaggio per comunicare che non è stato trasferito il numero di caratteri richiesto. La funzione ReadRec usa questa tecnica di verifica per ognuna delle cinque parti del record.

La visualizzazione

Se la funzione Read si svolge senza problemi (ovvero restituisce un valore ccount diverso da −1), il passo successivo è la visualizzazione del record nella finestra Intuition, usando la funzione di programma ConsolePrint (si ricordi che ogni stringa dev'essere a terminazione nulla). Il programma, per la precisione, ricorre a tre serie di tre chiamate a ConsolePrint per collocare nella finestra testo le diverse parti del record.

Dopo che tutte le cinque parti del record sono state memorizzate nelle variabili locali ReadRec, vengono visualizzate nella finestra; il programma ricorre quindi alla funzione AllocMem della libreria Exec per allocare la memoria necessaria per un esemplare di struttura DBList (definita tra i dati globali nella prima parte del programma).

Si noti che il programma controlla il valore restituito del parametro nextp, per verificare che ci sia abbastanza memoria per allocare il nuovo esemplare di struttura DBList.

Se l'esecuzione di AllocMem si conclude senza intoppi, il programma esegue tre chiamate alla funzione "strcpy" per copiare le singole parti del record negli appropriati parametri della struttura DBList e quindi li visualizza chiamando tre volte ConsolePrint. Queste istruzioni di visualizzazione servono a verificare che i dati del record siano stati collocati al posto giusto all'interno della struttura DBList.

Si noti che ReadRec ha semplicemente lo scopo di leggere i dati del record dal file MioDatabase. Per non complicare troppo le cose, non abbiamo collegato i record in una lista concatenata. Comunque questo è uno degli ampliamenti al programma che ci proponiamo di realizzare in una successiva versione.

Nel prossimo numero

Con il listato che segue, siamo andati un po' più vicini a un vero (e utile) programma stile-Intuition: abbiamo aggiunto un importante menu e abbiamo collegato la finestra testo al dispositivo Console, per utilizzarlo al fine di visualizzare testi provenienti da *qualunque* sorgente. Nel caso preso in esame, il testo veniva fornito da un file preesistente. Nel prossimo numero vedremo come si possono introdurre caratteri digitati da tastiera. In tal modo, l'utente potrà inserire dati per nuovi record, ed estendere così il database già esistente (come nel programma *Rubrica*). Inoltre, i caratteri digitati dall'utente per essere inseriti nel database potranno essere visualizzati nella finestra (il cosiddetto "echo") in modo che l'utente possa verificarne la correttezza.

Un'altra caratteristica del prossimo programma sarà una sequenza di istruzioni per creare e gestire dinamicamente una lista. Affronteremo così la creazione e la gestione delle liste in linguaggio C e, parallelamente, in *F-Basic*.

Listato: ***Testo***

```
/* FILE INCLUDE */

#include "libraries/dos.h"
#include "libraries/dosextens.h"

#include "stdio.h"

#include "exec/exec.h"
#include "exec/types.h"
#include "exec/io.h"
#include "exec/memory.h"

#include "intuition/intuition.h"
#include "intuition/intuitionbase.h"

/* Definizioni */

#define WINDOWFLAGS (SMART_REFRESH | REPORTMOUSE |
                 WINDOWSIZING | WINDOWDRAG |
                   WINDOWDEPTH | WINDOWCLOSE)
#define IDCMPFLAGS (RAWKEY | ACTIVEWINDOW |
                  INACTIVEWINDOW | CLOSEWINDOW |
                 REFRESHWINDOW | MENUPICK |
                  MOUSEBUTTONS | MOUSEMOVE )

#define MENUITEMFLAGS (ITEMTEXT | ITEMENABLED |
                   HIGHCOMP)
#define MENUFLAGS MENUENABLED
#define MESSAGGIO1
```

```
    (1L << MyWindow1->UserPort->mp_SigBit)

/* Variabili globali */

struct FileHandle *Open();
ULONG Read();

struct FileHandle *fh  = 0L;
char *filename = "MioDatabase";

/* Stringa di preparazione per la console */

char ConsolePrep[] =
{
  0x9B, 0x35, 0x35, 0x79,
  0x9B, 0x30, 0x20, 0x70, 0x00
};

extern struct MsgPort *CreatePort();
extern struct IOStdReq *CreateStdIO();

void *OpenLibrary();
struct Window *OpenWindow();
struct IntuiMessage *GetMsg();

struct MsgPort *MyMsgPort;
struct IOStdReq *MyRequest;

struct Screen *MyScreen;
struct Window *MyWindow1;

struct IntuitionBase *IntuitionBase;
struct IntuiMessage *MyMenuMessage;

struct IntuiText MyMenuItemText0 =
```

```
 {1, 2, JAM2, 0, 0, NULL, "Clear", NULL };

struct IntuiText MyMenuItemText1 =
  { 1, 2, JAM2, 0, 0, NULL, "Open", NULL };

struct IntuiText MyMenuItemText2 =
  { 1, 2, JAM2, 0, 0, NULL, "Save", NULL };

struct IntuiText MyMenuItemText3 =
  { 1, 2, JAM2, 0, 0, NULL, "Save AS", NULL };

struct IntuiText MyMenuItemText4 =
  { 1, 2, JAM2, 0, 0, NULL, "Quit", NULL };

/* definizioni testi per le voci del menu User */

struct IntuiText MyMenuItemText5 =
  { 1, 2, JAM2, 0, 0, NULL, "Read Record File",
    NULL };

struct IntuiText MyMenuItemText6 =
  { 1, 2, JAM2, 0, 0, NULL, "Write Record File",
    NULL };

struct IntuiText MyMenuItemText7 =
  { 1, 2, JAM2, 0, 0, NULL, "Add Record", NULL };

struct IntuiText MyMenuItemText8 =
  { 1, 2, JAM2, 0, 0, NULL, "Delete Record", NULL }
;

struct IntuiText MyMenuItemText9 =
  { 1, 2, JAM2, 0, 0, NULL, "Erase All Records",
    NULL };

struct IntuiText MyMenuItemText10 =
  { 1, 2, JAM2, 0, 0, NULL, "List All Records",
    NULL };

struct MenuItem MyMenuItem4 =
  { NULL, 0, 125, 60, 10, MENUITEMFLAGS, 0,
    (APTR)&MyMenuItemText4,NULL, NULL, NULL,
    MENUNULL };

struct MenuItem MyMenuItem3 =
  { &MyMenuItem4, 0, 104, 60, 10, MENUITEMFLAGS,
    0, (APTR)&MyMenuItemText3,
    NULL, NULL, NULL, MENUNULL };

struct MenuItem MyMenuItem2 =
  { &MyMenuItem3, 0, 83, 60, 10, MENUITEMFLAGS,
    0, (APTR)&MyMenuItemText2,
    NULL, NULL, NULL, MENUNULL };

struct MenuItem MyMenuItem1 =
  { &MyMenuItem2, 0, 42, 60, 10, MENUITEMFLAGS,
    0, (APTR)&MyMenuItemText1,
    NULL, NULL, NULL, MENUNULL };

struct MenuItem MyMenuItem0 =
  { &MyMenuItem1, 0, 19, 60, 10, MENUITEMFLAGS,
    0, (APTR)&MyMenuItemText0,
    NULL, NULL, NULL, MENUNULL };

/* definizione delle voci del menu User */

struct MenuItem MyMenuItem10 =
  { NULL, 0, 125, 150, 10, MENUITEMFLAGS, 0, (APTR)
   &MyMenuItemText10, NULL, NULL, NULL, MENUNULL };

struct MenuItem MyMenuItem9 =
  { &MyMenuItem10, 0, 105, 150, 10, MENUITEMFLAGS,
    0, (APTR)&MyMenuItemText9,
    NULL, NULL, NULL, MENUNULL };

struct MenuItem MyMenuItem8 =
  { &MyMenuItem9, 0, 85, 150, 10, MENUITEMFLAGS,
    0, (APTR)&MyMenuItemText8,
    NULL, NULL, NULL, MENUNULL };

struct MenuItem MyMenuItem7 =
  { &MyMenuItem8, 0, 65, 150, 10, MENUITEMFLAGS,
    0, (APTR)&MyMenuItemText7,
    NULL, NULL, NULL, MENUNULL };

struct MenuItem MyMenuItem6 =
  { &MyMenuItem7, 0, 45, 150, 10, MENUITEMFLAGS,
    0, (APTR)&MyMenuItemText6,
    NULL, NULL, NULL, MENUNULL };

struct MenuItem MyMenuItem5 =
  { &MyMenuItem6, 0, 25, 150, 10, MENUITEMFLAGS,
    0, (APTR)&MyMenuItemText5,
    NULL, NULL, NULL, MENUNULL };

struct Menu MyMenu5 =
  { NULL, 310, 0, 50, 10, MENUFLAGS, "User",
    &MyMenuItem5 };

struct Menu MyMenu4 =
  { &MyMenu5, 250, 0, 50, 10, MENUFLAGS, "Random",
    NULL };

struct Menu MyMenu3 =
  { &MyMenu4, 190, 0, 50, 10, MENUFLAGS, "Search",
    NULL };

struct Menu MyMenu2 =
  { &MyMenu3, 130, 0, 50, 10, MENUFLAGS, "Cursor",
    NULL };

struct Menu MyMenu1 =
  { &MyMenu2, 70, 0, 50, 10, MENUFLAGS, "Edit",
    NULL };

struct Menu MyMenu0 =
  { &MyMenu1, 0, 0, 60, 10, MENUFLAGS, "Project",
    &MyMenuItem0 };

struct NewScreen MyNewScreen =
  { 0, 0, 640, 200, 4, 1, 2, HIRES, CUSTOMSCREEN,
    NULL, "MioSchermo", NULL, NULL };

struct NewWindow MyNewWindow1 =
  { 20, 20, 600, 160, 1, 2, IDCMPFLAGS,
    ( ACTIVATE  WINDOWFLAGS ), NULL, NULL,
    "MiaFinestraTesto", NULL, NULL, 50, 50,
    640, 200, CUSTOMSCREEN };

/* definizione della struttura DBLIST  */

struct DBLIST {
  char data;
  char name[10];
  char phone[10];
  struct DBLIST *link;
};

/* definizione di due puntatori alla struttura
   DBLIST */

struct DBLIST *head;
struct DBLIST *nextp;

LONG mask = 0;
#define INTUITION 0x00000001
#define SCREEN    0x00000004
#define WINDOW1   0x00000008
#define MSGPORT   0x00000016
```

```
#define REQUEST   0x00000032
#define DEVICE    0x00000064

#define PROJECT 0
#define EDIT    1
#define CURSOR  2
#define SEARCH  3
#define RANDOM  4
#define USER    5

#define CLEAR   0
#define OPEN    1
#define SAVE    2
#define SAVEAS  3
#define QUIT    4

/* definizione dei numeri delle voci del menu
   User */

#define READ    0
#define WRITE   1
#define ADD     2
#define DELETE  3
#define LIST    4
#define ERASE   5

/* ****************************************** */

main()
{
  ULONG classe;
  UWORD codice;

  if( !(IntuitionBase = (struct IntuitionBase *)
        OpenLibrary("intuition.library", 0)))
  {
    printf("Niente libreria Intuition\n");
    CloseAll();
    exit(2);
  }

  mask| = INTUITION;

XXX  if( !(MyScreen = (struct Screen *)
        OpenScreen(&MyNewScreen)))
  {
    printf("Impossibile aprire lo schermo\n");
    CloseAll();
    exit(3);
  }

  mask| = SCREEN;

  MyNewWindow1.Screen = MyScreen;

  if( !(MyWindow1 = (struct Window *)
        OpenWindow(&MyNewWindow1)))
  {
    printf("Impossibile aprire la finestra\n");
    CloseAll();
    exit(4);
  }

  mask| = WINDOW1;

 /* Allocazione di una message port per le
    comunicazioni con il dispositivo Console */

  if (!(MyMsgPort = (struct MsgPort *)
        CreatePort(0,0)))
  {
    printf("Impossibile creare la message port\n")
;
    CloseAll();
    exit(5);
  }

  mask| = MSGPORT;

 /* Allocazione di una struttura IOStdReq per le
    comunicazioni con il dispositivo Console */

  if (!(MyRequest = (struct IOStdReq *)
        CreateStdIO(MyMsgPort)))
  {
    printf(
    "Impossibile creare la struttura IOStdReq\n"
          );
    CloseAll();
    exit(6);
  }

  mask| = REQUEST;

 /* Inizializzazione della struttura IOStdReq */

  MyRequest->io_Data   = (APTR) MyWindow1;
  MyRequest->io_Length = sizeof(*MyWindow1);

 /* Apertura della console */

  if (OpenDevice("console.device", 0L,
      MyRequest, 0L))
  {
    printf("Impossibile aprire la console\n");
    CloseAll();
    exit(7);
  }

  mask| = DEVICE;

 /* Inizializzazione del dispositivo Console */

  ConsolePrint(MyRequest, ConsolePrep);

  ConsolePrint(MyRequest, "Messaggio stampato
  usando il dispositivo Console\n");

  MyScreen->FirstWindow = MyWindow1;

  MyWindow1->NextWindow = NULL;

  SetMenuStrip( MyWindow1, &MyMenu0);
  OnMenu( MyWindow1, NOITEM);

 /* Loop di gestione dei messaggi */

  for (;;)
  {
    if( (MyMenuMessage = (struct IntuiMessage *)
         GetMsg(MyWindow1->UserPort)) == 0L)
    {
      Wait(MESSAGGIO1);
      continue;
    }

    classe = MyMenuMessage->Class;
    codice = MyMenuMessage->Code;
    ReplyMsg(MyMenuMessage);

    switch (classe)
    {
      case CLOSEWINDOW :  CloseAll();
                          exit(0);
                          break;

      case MENUPICK :     if( MENUNUM(codice) !=
                              MENUNULL)
```

```
                        ProcessMenuItems(
                          MyWindow1,
                          MENUNUM(codice),
                          ITEMNUM(codice),
                          SUBNUM(codice));
                        break;

      case MOUSEBUTTONS   : break;
      case ACTIVEWINDOW   : break;
      case INACTIVEWINDOW : break;

    }
  }
} /* Fine del main */

/* ***************************************** */

ProcessMenuItems(w, menu, voce, sotto-voce)
struct Window *w;
int menu;
int voce;
int sotto-voce;
{

 switch(menu)
 {
    case PROJECT:

      switch(voce)
      {
        case CLEAR:  ConsolePrint(MyRequest,
                     "Sto ripulendo il buffer\n");

                     ClearBuffer();
                     break;

        case OPEN:   if(fh != 0L)
                     ConsolePrint(MyRequest,
                     "Database gia' aperto\n");
                     if(fh == 0L)
                     {
                       ConsolePrint(MyRequest,
                       "Database aperto\n");
                       OpenFile(filename);
                     }
                     break;

        case SAVE:   if(fh != 0L)
                     {
                       ConsolePrint(MyRequest,
                       "Database salvato\n");
                       SaveFile(filename);
                     }
                     if(fh == 0L)
                     ConsolePrint(MyRequest,
                     "Database non aperto\n");
                     break;

        case SAVEAS: if(fh != 0L)
                     {
                       ConsolePrint(MyRequest,
                       "Database salvato\n");
                       SaveFile(filename);
                     }
                     if(fh == 0L)
                     ConsolePrint(MyRequest,
                     "Database non aperto\n");
                     break;

        case QUIT:   ConsolePrint(MyRequest,
                     "Uscita dal programma\n");
                    CloseAll();
                    break;
      default:
        break;
     }
    break;

   case USER:

    switch(voce)
    {
      case READ:   if(fh != 0L)
                   {
                     ConsolePrint(MyRequest,
                    "Lettura del database\n");
                     ReadRec(filename);
                   }
                   if(fh == 0L)
                   ConsolePrint(MyRequest,
                   "Database non aperto\n");
                   break;

      case WRITE:  if(fh != 0L)
                   {
                     ConsolePrint(MyRequest,
                     "Scrittura del database\n");
                     WriteRec(filename);
                   }
                   if(fh == 0L)
                   ConsolePrint(MyRequest,
                   "Database non aperto\n");
                   break;

      case ADD:    ConsolePrint(MyRequest,
                   "Aggiunta di un record\n");
                   AddRec();
                   break;

      case DELETE: ConsolePrint(MyRequest,
                   "Eliminazione di un record\n");
                   DeleteRec();
                   break;

      case LIST:   ConsolePrint(MyRequest,
                   "Lista dei record\n");
                   ListRec();
                   break;

      case ERASE:  ConsolePrint(MyRequest,
                   "Cancellazione database\n");
                   EraseRec();
                   break;

      default:
      break;
    }

    default:
    break;
 }
} /* Fine di ProcessMenuItems */

/* ***************************************** */

CloseAll()
{
  ClearMenuStrip(MyWindow1, &MyMenu0);

  if( mask & WINDOW1 )  CloseWindow(MyWindow1);
  if( mask & SCREEN )   CloseScreen(MyScreen);
  if( mask & INTUITION)
     CloseLibrary(IntuitionBase);
  if( mask & MSGPORT)   DeletePort(MyMsgPort);
  if( mask & DEVICE)    CloseDevice(MyRequest);
  if( mask & REQUEST)   DeleteStdIO(MyRequest);
```

```
  if(fh != 0L) Close(fh);

  exit(0);
} /* Fine di CloseAll */

/* ****************************************** */

OpenFile(fn)
char *fn;
{
  head = NULL;

  if(fh == 0) fh = (struct FileHandle *)
     Open(fn, MODE_OLDFILE);

  if(fh == 0)
  {
    ConsolePrint(MyRequest,
    "Il database non puo' essere aperto\n");
    ConsolePrint(MyRequest,
    "Trasferire il file nella directory\n");
    return;
  }

  ConsolePrint(MyRequest,
  "Il database e' stato aperto\n");
  ConsolePrint(MyRequest,
  "Leggere i record tramite l'opzione READ\n");

} /* Fine di OpenFile */

/* ****************************************** */

ReadRec(fn)
char *fn;
{
  int ccount;

  char k[2], c1[2], c2[2], r[2];
  char newname[11], newphone[11];

  ccount = Read(fh, &k, 1);
  if(ccount == -1)
  {
    ConsolePrint(MyRequest,
    "Errore leggendo il record\n");
    return;
  }

/* Termina la stringa k con un NULL */

  *(k + 1) = '\0';

  ConsolePrint(MyRequest,
  "Il valore dell'indice e' ");
  ConsolePrint(MyRequest, &k);
  ConsolePrint(MyRequest, "\n");

  ccount = Read(fh, &c1, 1);
  if(ccount == -1)
  {
    ConsolePrint(MyRequest,
    "Errore nella lettura della virgola\n");
    return;
  }

  ccount = Read(fh, &newname, 10);
  if(ccount == -1)
  {
    ConsolePrint(MyRequest,
    "Errore nella lettura del nome\n");
    return;
  }

/* Termina la stringa newname con un NULL */

  *(newname + 10)= '\0';

  ConsolePrint(MyRequest,
  "Il nome e' ");
  ConsolePrint(MyRequest, &newname);
  ConsolePrint(MyRequest, "\n");

  ccount = Read(fh, &c2, 1);
  if(ccount == -1)
  {
    ConsolePrint(MyRequest,
    "Errore nella lettura della virgola\n");
    return;
  }

  ccount = Read(fh, &newphone, 10);
  if(ccount == -1)
  {
    ConsolePrint(MyRequest,
    "Errore nella lettura del telefono\n");
    return;
  }

/* Termina la stringa newphone con un NULL */

  *(newphone + 10)= '\0';

  ConsolePrint(MyRequest,
  "Il telefono e' ");
  ConsolePrint(MyRequest, &newphone);
  ConsolePrint(MyRequest, "\n");

  ccount = Read(fh, &r, 1);
  if(ccount == -1)
  {
    ConsolePrint(MyRequest,
    "Errore nella lettura del RETURN\n");
    return;
  }

/* Allocazione di una nuova struttura DBLIST */

  nextp = (struct DBLIST *)
  AllocMem(sizeof(struct DBLIST), MEMF_CHIP);

/* Copia le variabili locali nella struttura */

  strcpy(nextp->data, &k);
  ConsolePrint(MyRequest,
  "Il valore di nextp->data e' ");
  ConsolePrint(MyRequest, nextp->data);
  ConsolePrint(MyRequest, "\n");

  strcpy(nextp->name, &newname);
  ConsolePrint(MyRequest,
  "Il valore di nextp->name e' ");
  ConsolePrint(MyRequest, nextp->name);
  ConsolePrint(MyRequest, "\n");

  strcpy(nextp->phone, &newphone);
  ConsolePrint(MyRequest,
  "Il valore di nextp->phone e' ");
  ConsolePrint(MyRequest, nextp->phone);
  ConsolePrint(MyRequest, "\n");

} /* Fine di ReadRec */

/* ****************************************** */
```

```
ConsolePrint(IOSR, string)
struct IOStdReq *IOSR;
char *string;
{
  if(*string)
  {
    IOSR->io_Data    = (APTR) string;
    IOSR->io_Length  = -1L;
    IOSR->io_Command = CMD_WRITE;

   (void) DoIO(IOSR);
  }
} /* Fine di ConsolePrint */

/* ****************************************** */

ClearBuffer()
{
}

/* ****************************************** */

SaveFile(fn)
char *fn;
{
}

/* ****************************************** */

WriteRec(fn)
char *fn;
{
}

/* ****************************************** */

AddRec()
{
}

/* ****************************************** */

DeleteRec()
{
}

/* ****************************************** */

ListRec()
{
}

/* ****************************************** */

EraseRec()
{
}
```

ABBONATI!

A CASA TUA UN REGALO OGNI MESE

ABBONARSI CONVIENE!

- Sicurezza di non perdere neanche un numero
- Prezzo bloccato per un anno
- Sconto del 15% sul prezzo di copertina

- Possibilità d'includere nell'abbonamento gli arretrati
- Comodità di ricevere la rivista direttamente a casa

Per chiunque sia interessato ai computer Commodore, *Commodore Gazette* è indispensabile. Nessuna rivista in Italia offre ai suoi lettori tanta qualità con recensioni hardware e software, listati, presentazioni esclusive, informazioni di ogni genere riguardanti C-64, C-128 e Amiga.

Un abbonamento a *Commodore Gazette* è il regalo più bello che possiate fare a voi stessi e agli altri... un regalo nuovo ogni mese. Ma non è finita! Risparmierete il 15% sul prezzo di copertina e potrete includere nel prezzo dell'abbonamento anche i numeri arretrati che mancano alla vostra raccolta. Resta inteso che per ogni arretrato scelto verrà spedito un numero di prossima uscita in meno. Per esempio, chi si abbona a 12 numeri a partire dal n. 1/89 e richiede 5 arretrati, riceverà 7 numeri del 1989 e 5 arretrati.

Ritagliare e spedire a: IHT Gruppo Editoriale – Via Monte Napoleone, 9 – 20121 Milano

Sì, desidero sottoscrivere un abbonamento a ☐ 12 numeri (L. 81.600) ☐ 24 numeri (L. 163.200) di COMMODORE GAZETTE usufruendo così dello sconto del 15% sul prezzo di copertina

Nome e cognome ....

Indirizzo .... Città .... CAP ....

Inizio abbonamento dal n. .... Desidero ricevere i seguenti numeri arretrati (specificare numero e anno) ....

☐ Allego assegno bancario o circolare ☐ Allego fotocopia della ricevuta del vaglia postale

Firma ....

UN DATABASE CON IL COMPILATORE F-BASIC

F-Basic è un sistema di programmazione Basic con caratteristiche "stile-C". Lo mettiamo ora alla prova con un semplice programma di rubrica telefonica, accuratamente analizzato e descritto nell'articolo

di Eugene P. Mortimore

In un precedente articolo (pubblicato nel numero 5/89 di *Commodore Gazette*) abbiamo descritto il sistema di programmazione *F-Basic*, un pacchetto che mette in grado di costruire complesse applicazioni composte di un modulo principale (main) e di un certo numero di sottomoduli, o segmenti, di codici eseguibili. Abbiamo sottolineato che *F-Basic* è un *compilatore* Basic, in grado di offrire un'eccellente velocità di esecuzione e altre caratteristiche che lo avvicinano molto ai sistemi di programmazione in linguaggio C.

Abbiamo osservato che molte caratteristiche di *F-Basic* hanno una controparte diretta nel C, e lavorare in questo ambiente di programmazione diventa quindi un ottimo punto di partenza per imparare il linguaggio C. Questo articolo si propone d'illustrare nei dettagli le principali affinità, fornendo un concreto esempio di programmazione che può anche diventare, con un certo lavoro di ampliamento e personalizzazione, un programma realmente utile.

Le principali affinità

Se ci sono aspetti comuni tra C e *F-Basic*, quest'ultimo deve possedere ovviamente la capacità di creare e gestire strutture. L'equivalente delle strutture con *F-Basic* sono i **record**, o strutture Record.

Proprio come le strutture del linguaggio C, le strutture Record permettono di riunire insieme dati dissimili in un elenco a cui viene dato un nome, e poi di dichiarare nel programma diversi esemplari (instance) di ogni tipo di struttura. *F-Basic* mette a disposizione le funzioni New_Rec() e Dispose_Rec() per consentire al programma di allocare e disallocare gli specifici esemplari di struttura, al fine di gestire la memoria nel modo più economico possibile.

In un modo che ricorda molto da vicino l'allocazione e la disallocazione di memoria in C, ciascuna chiamata alla funzione New_Rec() restituisce un puntatore all'indirizzo della nuova struttura Record allocata, e ogni chiamata a Dispose_Rec() libera la memoria precedentemente allocata per la struttura. Anche la sintassi relativa all'assegnazione dei parametri è simile a quella del C.

Uno delle prime nozioni da apprendere per programmare nel familiare ambiente Basic "simulando" la programmazione in linguaggio C, è la gestione dei parametri puntatore. Ogni variabile di *F-Basic* dichiarata come variabile PTR_TO può essere utilizzata per puntare a una struttura Record, come si vedrà più dettagliatamente nella discussione sul programma d'esempio.

Informazioni preliminari

Il miglior modo per crearsi un ambiente di programmazione funzionale è riunire sullo stesso disco tutte le informazioni del sistema *F-Basic*, inserire questo disco nel disk drive DF1: e utilizzare il disk drive DF0: per il disco di programmazione in C che abbiamo descritto nell'articolo "Un ambiente di programmazione efficiente" (*Commodore Gazette* numero 2/89). In tal modo, tutti i comandi "C" (come COPY, RENAME...) saranno contenuti nel disco DF0:. Il disco DF1: conterrà quindi esclusivamente informazioni *F-Basic*, ovvero la directory FInclude (che contiene i file INCLUDE relativi a *F-Basic*) e i seguenti file: FastError, FastLib, FastIff, FastLinkLib e il file eseguibile SLDB (source level debugger).

È possibile ottenere questa organizzazione semplicemente copiando in un disco nuovo i file appropriati, traendoli dai tre dischi di cui si compone il pacchetto *F-Basic* (System, Example e Debugger). In questo modo avremo in DF1: tutti i file necessari alla programmazione e rimarrà ancora spazio sufficiente per ospitare i file sorgente creati con *F-Basic* e tutti i file supplementari che verranno prodotti nel corso delle varie fasi di programmazione (i file "DBG", "LNM", "BIN").

Il listato *Rubrica* illustra molti aspetti della programmazione con *F-Basic*. Ha lo scopo di creare un database interattivo contenente nomi e numeri telefonici, e salvare questo database su disco per poterlo richiamare o aggiornare in qualunque momento. Il programma permette all'utente di leggere e visualizzare i dati immagazzinati nel database, di aggiungere nuove voci e di cancellare quelle non più utili. Serve però anche a illustrare alcune delle più interessanti caratteristiche del sistema *F-Basic* di elaborazione dinamica delle liste e, con un po' di pratica, potrebbe essere trasformato in un programma di gestione di database veramente efficiente.

Come si può vedere, il programma consiste in un solo modulo *F-Basic*, che inizia con un'istruzione PROGRAM e si conclude con un'istruzione END. Tutte le istruzioni che iniziano con un "?" sono commenti, l'equivalente dei commenti /* */ nei file sorgente in linguaggio C.

Vi sono due punti etichettati, nel programma: {Again} (poco dopo l'inizio) e {Delay} (vicino alla fine). C'è un'istruzione "GOTO Again" e ci sono parecchie istruzioni "GOSUB Delay" sparse qua e là nel corso del programma. L'opzione "F", inoltre, ha due punti etichettati che permettono un loop interno di salto nei codici stessi dell'opzione, come si vedrà meglio più avanti.

Appena sopra l'etichetta {Again} si trovano tutte le variabili locali *definite* (struttura List) e *dichiarate* per un uso successivo da parte del programma. Si possono anche notare parecchie istruzioni che definiscono come Record un nuovo tipo di variabile chiamato List. Sono abbastanza evidenti le somiglianze di List con una struttura "stile C" con tre semplici voci ("Data", "Name" e "Phone") e un puntatore ("Link").

Dal momento che la struttura Record di tipo List contiene un puntatore, essa fornisce al database un meccanismo per costruire una lista concatenata *dinamicamente* (dotata cioè di collegamenti che entrano in gioco durante l'esecuzione del programma, non durante la compilazione). La lista è inoltre di tipo singolo (non una lista a doppia concatenazione, che può essere percorsa in entrambe le direzioni); in altre parole, ogni nodo della lista punta soltanto al nodo successivo, e non è collegato al nodo precedente, cosa che richiederebbe un secondo puntatore.

Si noti che la definizione del record List inizia con la parola chiave TYPE e termina con l'altra parola chiave ENDTYPE. Subito dopo la definizione di List, il programma dichiara un insieme di variabili locali: tre puntatori ("HEAD", "NEXTP" e "TRAIL") che puntano alla variabile di tipo List appena definita. Per la precisione, ciascuno dei tre puntatori punta a un diverso esemplare (instance) del record List, e lo specifico esemplare può variare durante l'esecuzione del programma. Come vedremo più avanti, il buon funzionamento del programma dipende in massima parte dall'uso che viene fatto delle variabili puntatore delle strutture List.

Il programma

Il programma raggiunge i suoi obiettivi fornendo all'utente sette opzioni, che possono essere chiamate interattivamente e sequenzialmente, digitando un solo carattere (sulla base di un menu che compare al centro dello schermo). Le sette opzioni sono "A" (add, aggiungere), "D" (delete, cancellare un record), "E" (erase, cancellare la lista), "F" (file), "L" (list, elencare), "S" (save, salvare), "Q" (quit, abbandonare). Il programma fa uso del comando WHEN di *F-Basic* per controllare il valore di una variabile di tipo Text denominata STR (dichiarata come STR*1, per indicare che contiene un solo carattere) e per intraprendere un'azione particolare a seconda del valore assegnato dall'utente alla variabile STR.

Il programma dichiara poi altre tre variabili che in seguito userà per costruire i nodi delle liste concatenate quando l'utente fornirà le necessarie informazioni da tastiera, oppure le caricherà da disco.

Tutte e tre le variabili sono dichiarate come variabili Text: STR con un solo carattere, NAME e PHONE con dieci caratteri. Dieci è il numero massimo di caratteri che si possono inserire nelle variabili, ma l'utente può utilizzarne anche un numero inferiore. In seguito vedremo che il programma utilizza anche le variabili locali I e K per costruire i nodi delle strutture List.

Compiute queste operazioni preliminari, inizia il loop principale del programma (all'altezza dell'etichetta Again). Si noti che, due righe prima, la variabile HEAD è stata inizializzata al valore NIL; NIL è l'equivalente di NULL in *F-Basic*, e rappresenta quindi il valore zero. Il valore zero serve in genere per individuare la fine (o l'inizio) di una lista in un loop di elaborazione. Quest'assegnazione particolare è situata proprio in questo punto del programma per indicare che all'inizio la lista è vuota e che verrà riempita aggiungendo (ADD) i nodi uno per uno tramite informazioni da tastiera, oppure caricando (READ) da disco le informazioni sui nodi. Quando l'utente fa la sua prima selezione dal menu, il programma gli chiede se intende creare i nodi per la lista "aggiungendoli" da tastiera oppure "leggendo" dal disco un file contenente nodi definiti in precedenza.

Il dialogo fra programma e utente inizia automaticamente quando il programma visualizza la

richiesta d'inserire informazioni. Sullo schermo appaiono le parole: "Enter Command..." (impartisci un comando), seguite dalle iniziali delle sette opzioni. Dal listato si osserva che, prima della visualizzazione del messaggio, il cursore viene collocato all'ottava riga e all'undicesima colonna dello schermo. Terminato il messaggio, il cursore viene ricollocato nella stessa posizione e il programma provvede a esaminare il carattere digitato in risposta dall'utente.

Supponiamo che l'utente decida di aggiungere (ADD) interattivamente nuovi nodi alla lista, fornendo per ogni nodo le informazioni richieste dalla struttura List. Il programma impiega l'opzione WHEN STR IS all'inizio del loop principale, e da lì salta al punto che si accorda con la scelta fatta dall'utente (è una procedura che ricorda il loop di elaborazione del menu principale di Intuition).

Dal punto di vista del funzionamento, il meccanismo dei salti è simile a quello che si ottiene in ambiente C con l'istruzione SWITCH. Quando il programma entra nel segmento di codice dell'opzione "A", il cursore viene subito riportato nella posizione iniziale e un prompt chiede all'utente d'inserire un numero intero che viene assegnato alla variabile locale K.

Quindi il programma cancella lo schermo, riporta di nuovo il cursore al punto di partenza e invia un altro prompt all'utente, perché inserisca il testo che rappresenta il nome da aggiungere alla lista; l'informazione viene immagazzinata nella variabile locale NEWNAME. La stessa operazione viene ripetuta con la variabile locale NEWPHONE.

Quando l'utente ha fornito queste tre informazioni, il programma alloca un nuovo esemplare di struttura Record di tipo List; in memoria ora risiede un esemplare non inizializzato della struttura, il cui indirizzo è contenuto nel puntatore NEXTP. Le tre istruzioni successive del programma si limitano a copiare nei parametri della struttura List i tre valori immagazzinati nelle variabili locali, e a quel punto la struttura è inizializzata.

Il programma costruisce quindi il collegamento tra il nodo appena aggiunto e il precedente. Si noti che il parametro LINK del nuovo nodo viene eguagliato al valore della variabile HEAD (quella che all'inizio del programma era stata inizializzata a NIL): in altre parole, questo è il primo nodo della lista e quindi punta a un nodo vuoto.

Come passo successivo, alla variabile locale HEAD viene attribuito il valore della variabile NEXTP (ricordiamo che NEXTP è il valore restituito dalla chiamata alla funzione New_Rec(), ed è un puntatore che individua la struttura List appena aggiunta alla lista). Quindi, il risultato di quest'operazione è che la variabile HEAD ora punta all'ultimo nodo aggiunto, quello che contiene il nuovo nome e il nuovo numero telefonico.

Per riassumere: l'utente fornisce tre informazioni, che vengono dapprima inserite in variabili di programma, e successivamente copiate nei parametri di un esemplare di struttura List; quindi questa struttura List viene collegata a una lista di strutture dello stesso tipo.

Se l'utente seleziona di nuovo l'opzione "A", il programma ripercorre lo stesso loop, ma quando arriva all'istruzione di assegnazione del parametro LINK, trova che il valore della variabile HEAD non è NIL, ma l'indirizzo dell'ultimo nodo inserito nella lista. In tal modo, a mano a mano che si aggiungono nuovi nodi, il programma provvede automaticamente a fare in modo che la variabile HEAD punti sempre all'ultimo nodo aggiunto. La lista dei nodi è "ancorata" nella parte alta a un valore NIL.

L'opzione "L"

L'opzione LIST serve a "sfogliare" il database, dal momento che visualizza sullo schermo i dati contenuti nei vari nodi. Per prima cosa il programma cancella lo schermo, quindi imposta il colore di primo piano a 3 (arancione) e varia lo stile dei caratteri a 1 (neretto). Quindi muove il cursore alla riga 6, colonna 19 e stampa sullo schermo il titolo della colonna. Subito dopo cambia il colore di primo piano in 2 (nero). Imposta la variabile NEXTP al valore raggiunto dalla variabile HEAD e avvia un loop WHILE, il quale *procede all'indietro* dall'ultimo nodo aggiunto fino al primo nodo NIL (che costituisce sempre l'origine della lista).

Il programma colloca quindi il cursore nella posizione opportuna, e poi visualizza i dati del nodo. Il processo si ripete per ogni nodo della lista. Si noti che il programma "spegne" il cursore durante la visualizzazione dei nodi; suggeriamo di provare a togliere quest'istruzione, e stare a vedere quel che succede.

Quando la visualizzazione è finita, il programma torna allo stile 0, cosicché continuando l'esecuzione (o uscendo dal programma, se l'utente seleziona l'opzione Q) lo stile dei caratteri appare quello di sempre.

Si noti infine che il programma contiene l'istruzione DELAY(50), per assegnare alla lista un certo tempo di permanenza sullo schermo. Questo tempo può essere regolato come si preferisce.

L'opzione "F"

L'opzione F è un utilissimo sotto-programma che permette all'utente di salvare su disco il proprio database, e di richiamarlo in memoria quando è necessario.

Per prima cosa il programma apre un file di nome "MioDatabase" (si noti che *F-Basic* assegna al file il numero di unità logica 1). Subito dopo il programma imposta la variabile HEAD a NIL, a indicare che la lista in costruzione avrà un nodo di testa nullo (in altre parole, non esistono altri nodi da leggere prima di quello nullo).

Finite queste operazioni preliminari, il programma utilizza un loop READ per leggere i singoli

record del file su disco. A questo scopo si serve dell'etichetta {Cont1} e dell'istruzione "GOTO Cont1". Ogni istruzione INPUT# legge i tre valori K, NEWNAME e NEWPHONE di un record; quando i valori sono stati letti, il programma chiama la funzione New_Rec() per allocare una struttura List. Come avviene con l'opzione ADD, il programma restituisce nella variabile locale NEXTP un puntatore alla struttura List. Ora che ha a disposizione un puntatore, con le tre istruzioni successive il programma inizializza i tre parametri di struttura. Quindi il puntatore LINK viene inizializzato per puntare al valore del parametro HEAD, e infine il parametro HEAD viene aggiornato con il valore appena restituito da NEXTP. Fatto ciò, il programma salta all'etichetta {Cont1} per leggere il record successivo.

Questo processo di lettura circolare si interrompe quando il programma incontra nel file un carattere EOF (fine del file), e salta all'etichetta {Cont2}. L'uscita dal loop si ottiene con l'opzione "EOF=" nell'istruzione INPUT#. Per la descrizione di questa opzione vi rimandiamo al manuale del programma.

Quando il loop di lettura da disco è concluso, tutti i record del file sono stati inseriti nelle strutture List, le quali sono state opportunamente collegate tra loro. Si può verificare la riuscita dell'operazione selezionando l'opzione "L" subito dopo l'opzione "F", per controllare se tutte le informazioni presenti nel file su disco si trovano nella lista di strutture List.

Le altre opzioni

Le altre opzioni funzionano in modo simile, utilizzando opportunamente i puntatori. Si noti che la lista varia in continuazione: l'utente può *add* (aggiungere) nodi, può *delete* (cancellarli), *erase* (cancellare il contenuto dell'intera lista) e *read* (copiare nodi dal disco alla memoria).

Il programma assicura che la variabile locale HEAD punti sempre all'ultimo nodo (in qualunque modo sia stato aggiunto). Una lista può quindi essere considerata come un elenco scritto su un foglio di carta, costruito partendo dall'alto e scendendo verso il fondo della pagina.

L'opzione "Q" si limita a chiamare il comando Basic STOP per uscire dal programma; non è necessario chiudere il file, dal momento che i loop "F" o "S" lo chiudono sempre impartendo al termine dell'esecuzione un'istruzione CLOSE(1).

In linea generale, il funzionamento del comando STOP è analogo a quello dell'istruzione EXIT del C, che in genere viene utilizzata per uscire dal programma in punti specifici, a seconda delle condizioni da esso incontrate durante l'esecuzione.

Per quanto riguarda il funzionamento dal punto di vista dell'utente, è ovvio che quando si utilizza il programma per la prima volta non esiste alcun file LIST sul disco. Di conseguenza, quando il programma presenta il primo prompt, si deve creare una lista di nodi impiegando l'opzione "A". Dopo averne inseriti un certo numero, conviene salvarli in un file impiegando l'opzione "S".

Si continua quindi ad aggiungere (ADD) e salvare (SAVE) finché non si giudica conclusa la prima sessione di lavoro. A quel punto conviene usare l'opzione LIST (per l'ultima volta) per riesaminare tutti i nodi e controllarli; se qualcuno dev'essere corretto, si usa l'opzione DELETE per cancellarlo e l'opzione ADD per sostituirlo con il nodo corretto. Quando la lista ha raggiunto uno stato soddisfacente, la si salva (SAVE) sul disco per l'ultima volta.

Miglioramenti e compilazione

Come si è visto, i nodi possono essere corretti soltanto cancellandoli e rimpiazzandoli con altri, e questa non è ovviamente la soluzione migliore. Sarebbe necessaria un'opzione "M" (MODIFY, modificare) per correggere i dati. Si potrebbe anche aggiungere un'opzione SEARCH (cercare) che esamini la lista alla ricerca di nodi particolari, e un'opzione di paginazione per la visualizzazione del database durante il LIST.

Alla sessione successiva, si può chiamare l'opzione "F" per leggere il file immagazzinato sul disco e usare poi le opzioni "A" e "M" per aggiungere nuovi nodi e ampliare ulteriormente il database. Nel corso del processo può essere utile l'opzione SEARCH per controllare quali sono i dati già inseriti. Non appena la lista supera il contenuto di 15 nodi, l'opzione di paginazione permette di esaminare il database sullo schermo a gruppi di nodi (pagine).

Supponiamo che il vostro file sorgente *F-Basic* si chiami "rubrica". Per la compilazione è sufficiente la semplice istruzione:

```
FB rubrica
```

che produrrà la creazione del file "rubrica.bin", eseguibile direttamente da CLI digitando "rubrica.bin".

Si osservi che questa istruzione di compilazione si può utilizzare soltanto se il file eseguibile non supera i 50 mila byte. Se *F-Basic* reagisce con il messaggio INADEQUATE MEMORY (memoria inadeguata), al suo posto si può impiegare:

```
FB rubrica opt - o5000
```

Se si ottiene ancora il messaggio INADEQUATE MEMORY e si possiede un RAM disk contenente alcuni altri file, oppure un insieme di comandi residenti (caricati per esempio tramite il comando RESIDENT dell'AmigaDOS), si deve rimuovere dalla memoria il RAM disk o alcuni dei comandi residenti e ritentare la compilazione. Il problema dovrebbe presentarsi soltanto con macchine da 512K.

Nel prossimo articolo

Nel prossimo articolo descriveremo l'assemblatore *CAPE68K*. Le sue principali caratteristiche sono un'integrazione tra assemblatore e editor, con menu in stile Intuition e comandi equivalenti, nonché l'assemblaggio automatico da buffer di testo, con un'opzione di STOP ON ERROR. *CAPE68K* offre inoltre la possibilità di un numero illimitato di moduli di programma; l'uso di file INCLUDE preassemblati; un output compatibile con il disassemblatore *DSM* della OTG; un'interfaccia *AREXX* con la possibilità di macro; ottime caratteristiche di controllo dei codici oggetto; un'eccellente compatibilità con il macro assembler standard dell'Amiga.

Listato: *Rubrica in F-Basic*

```
PROGRAM DataBase
?
? Il programma illustra le strutture RECORD
? di F-Basic (liste a concatenazione semplice).
? Assicuratevi di utilizzare l'opzione "E"
? per cancellare la lista e di restituire
? al sistema la memoria impiegata
? PRIMA di uscire dal programma.
?
? Si definisce un nuovo tipo di "struttura"
? Record, denominata LIST
TYPE LIST IS RECORD
   INTEGER DATA
   TEXT NAME*10
   TEXT PHONE*10
   PTR_TO LIST LINK
ENDTYPE
? Si dichiarano alcune variabili puntatore
PTR_TO LIST HEAD,NEXTP,TRAIL
? Si dichiarano alcune variabili locali
? di tipo testo
TEXT STR*1
TEXT NEWNAME*10
TEXT NEWPHONE*10
? Si dichiarano due variabili locali
? di tipo indice
INTEGER K,I
? Si inizializza a NIL (NULL) la variabile HEAD
HEAD=NIL
? Inizia il loop principale
{Again}
  SCR_CLER
  CURS_LOC(8,11)
  INPUT "I comandi disponibili sono A=Add D=Delet
e E=Erase F=File L=List S=Save Q=Quit" STR
  CURS_LOC(8,11)
  PRINT SPACE(65)
? Si considerano le varie possibilita'
? di risposta dell'utente
  WHEN STR IS
? Elaborazione dell'opzione ADD
    ["A","a"]
        CURS_LOC(8,11)
        INPUT "Inserire un numero intero" K
        SCR_CLER
        CURS_LOC(8,11)
        INPUT "Inserire il nome" NEWNAME
        SCR_CLER
        CURS_LOC(8,11)
        INPUT "Inserire il numero telefonico" NE
            WPHONE
        NEXTP=NEW_REC(LIST)
        NEXTP.DATA=K
        NEXTP.NAME=NEWNAME
        NEXTP.PHONE=NEWPHONE
        NEXTP.LINK=HEAD
        HEAD=NEXTP
        GOSUB Delay
? Elaborazione dell'opzione DELETE
    ["D","d"]
        SCR_CLER
        CURS_LOC(8,11)
        INPUT "Inserire il numero della voce
            da cancellare" K
        TRAIL=HEAD
        NEXTP=HEAD
        WHILE NEXTP<>NIL AND NEXTP.DATA<>K DO
           TRAIL=NEXTP
           NEXTP=NEXTP.LINK
        ENDWHILE
        IF NEXTP=HEAD AND HEAD<>NIL THEN HEAD=T
           RAIL.LINK
        IF NEXTP<>NIL THEN
           TRAIL.LINK=NEXTP.LINK
           DISPOSE_REC(NEXTP)
           CURS_LOC(8,11)
           PRINT K[10]," E' stata cancellata"
        ELSE
           CURS_LOC(8,11)
           PRINT K[10]," Non e' stata trovata"
        ENDIF
        GOSUB Delay
? Elaborazione dell'opzione ERASE
    ["E","e"]
        IF (HEAD<>NIL) THEN
           REPEAT
              K=HEAD
              NEXTP=HEAD.LINK
              DISPOSE_REC(HEAD)
              HEAD=NEXTP
           UNTIL NIL=HEAD
        ENDIF
        CURS_LOC(8,11) ; PRINT "La lista e'
           stata cancellata... adesso e' vuota"
        GOSUB Delay
? Elaborazione dell'opzione FILE
    ["F","f"]
  OPEN(1, "MioDatabase")
  HEAD = NIL

{Cont1}
        INPUT# 1 EOF=Cont2 K, NEWNAME, NEWPHONE
```

```
        NEXTP=NEW_REC(LIST)
        NEXTP.DATA=K
        NEXTP.NAME=NEWNAME
        NEXTP.PHONE=NEWPHONE
        NEXTP.LINK=HEAD
        HEAD=NEXTP
   GOTO Cont1

{Cont2}
   CLOSE(1)
        GOSUB Delay
? Elaborazione dell'opzione LIST
     ["L","l"]
   CURS_INV
        SCR_CLER
   SCR_FORE(3)
   SCR_STYL(1)
        CURS_LOC(6, 19)
   PRINT "Numero Nome             Numero telefonico"
   SCR_FORE(2)
        NEXTP=HEAD
        WHILE NEXTP<>NIL DO
     CURS_LOC(NEXTP.DATA + 8, 8)
          PRINT NEXTP.DATA, SPACE(6), NEXTP.NAM
             E, SPACE(4), NEXTP.PHONE
          NEXTP=NEXTP.LINK
        ENDWHILE
   SCR_STYL(0)
   CURS_VIS
     DELAY(50)
          GOSUB Delay
? Elaborazione dell'opzione SAVE
     ["S","s"]
   OPENA(1, "MioDatabase")
          NEXTP=HEAD
          WHILE NEXTP<>NIL DO
             PRINT# 1 NEXTP.DATA, NEXTP.NAME, NEXT
P.PHONE
             NEXTP=NEXTP.LINK
          ENDWHILE
   CLOSE(1)
? Elaborazione dell'opzione QUIT
     ["Q","q"]
   STOP
? Elaborazione di risposte non corrette
     OTHERWISE
          CURS_LOC(8,11)
          PRINT "Comando non valido"
          GOSUB Delay
     ENDCASES
? Si torna all'inizio del loop di elaborazione
? delle opzioni
     GOTO Again
? Pausa di 30 tick
{Delay}
     DELAY(30)
     LRETURN
END
```

PRIMI PASSI CON L'AMIGA

L'EDITOR MEMACS IN DIECI COMANDI

Non spaventatevi per l'enormità di Memacs: l'elaborato editor del disco Extras può essere "domato" con dieci soli comandi

di Bill Catchings e Mark L. Van Name

Per chi non ha troppi soldi da spendere, il disco Extras 1.3 offre un efficace editor di testi, adatto tanto a scrivere il listato di un programma quanto a preparare una lettera per la zia. E ha un ottimo prezzo: è gratis. Si trova nella directory Tools, si chiama **Memacs** ed è una versione con mouse e menu del mastodontico word processor del MIT, *Emacs*. Qualunque funzione di edit vi venga in mente, anche la più bizzarra, *Emacs* ce l'ha. Non conosciamo nessun altro editor, per esempio, che abbia un comando per scambiare tra loro due caratteri (come si farebbe altrimenti a trasformare "gil" o "lgi" in "gli"?). Ma c'è un inconveniente: *Emacs* (come la sua precedente controparte per microcomputer, *MicroEmacs*), costringe a imparare speciali sequenze di caratteri di controllo per ogni comando... una litania non particolarmente divertente da ricordare a memoria.

Grazie ai menu, con *Memacs* non è più necessario imparare tutto a memoria, e grazie ai tasti freccia e al mouse è più facile muovere il cursore all'interno del testo. Però questi aiuti costringono l'utente ad abbandonare la tastiera e a impugnare il mouse praticamente per qualunque correzione al testo, il che può costituire una spiacevole interruzione del lavoro di battitura (almeno per gli utenti particolarmente a loro agio con la tastiera). Come compromesso all'affidarsi completamente all'uno o all'altro sistema, conviene memorizzare dieci comandi essenziali, e utilizzare i menu per il resto.

Per prima cosa, avviate *Memacs* con una doppia selezione sulla sua icona nello schermo Workbench, oppure digitando da una finestra CLI (dopo il prompt Extras 1.3:) il comando:

```
tools/MEMACS nome del file
```

Se siete già anche nella directory tools, basta digitare:

```
MEMACS nome del file
```

Quando *Memacs* va in esecuzione, appare una finestra-testo che occupa l'intero schermo, qualunque sia la grandezza dello schermo Workbench o della finestra CLI da cui il programma è stato chiamato. Sulla sommità della finestra *Memacs* c'è una tipica barra menu; sul fondo è presente invece una riga speciale, chiamata "riga-modo". *Memacs* usa questa riga per visualizzare informazioni relative alla finestra di lavoro. I contenuti della riga-modo vanno dalle informazioni standard, come il nome del file di lavoro, oppure la segnalazione che il file è stato modificato (rappresentata da un asterisco come secondo carattere), fino a prompt provvisori e richieste d'inserire informazioni, come nel caso di una stringa di ricerca.

I comandi essenziali

Memacs mette a disposizione due tipi di comandi: le combinazioni Ctrl e le metasequenze.

Le combinazioni Control si ottengono premendo il tasto Control in combinazione con un altro tasto (per brevità, sia nei menu sia nella documentazione, questa combinazione viene indicata con il carattere ^ seguito da un altro carattere (per esempio ^C significa Control/C).

Le metasequenze (così chiamate perché sui terminali del MIT venivano ottenute con tasti che si chiamavano "meta") si ottengono premendo prima il carattere Esc e poi un altro carattere. La documentazione di *Memacs* usa la notazione <ESC> seguita dalla lettera che rappresenta il comando, come per esempio <ESC>f o <ESC>b (nei menu c'è una differenza: al posto di <ESC>f si legge ESCf).

Per aiutarvi a ricordare a che cosa serve ciascun comando, tenete a mente che in genere la lettera è l'iniziale del suo nome. Come ulteriore aiuto, si

ricordi che le combinazioni Control e quelle meta con la stessa lettera sono spesso correlate. Sapendo qual è la funzione dell'una, spesso si riesce a indovinare anche la funzione dell'altra. Per un riferimento immediato, si vedano le tavole che accompagnano l'articolo.

Cominciamo ora a delineare l'insieme dei nostri comandi essenziali. Per eseguire una correzione, si deve innanzitutto sapere come muovere il cursore all'interno del testo; oltre al mouse e ai tasti freccia, vi sono parecchi comandi che fanno "saltare" il cursore in varie posizioni del testo. Per esempio il comando Next Line (^N) muove il cursore alla riga successiva. Il suo inverso, il comando Previous Line, è ^P. Allo stesso modo ci si può muovere in avanti (Forward) di un carattere utilizzando ^F, e indietro (Backward) utilizzando ^B.

Come avviene nella maggior parte dei word processor, se un comando porta il cursore in un punto del testo non visibile sullo schermo, il programma esegue automaticamente lo scroll delle pagine. Si può anche ordinare a *Memacs* di visualizzare la pagina successiva digitando ^V (comando View Next Page).

Vediamo ora un esempio delle realazioni tra combinazioni Control e meta. Dal momento che ^F e ^B muovono avanti e indietro di un carattere il cursore, è abbastanza facile imprimersi nella memoria che <ESC>f muove il cursore in avanti di una parola e <ESC>b indietro di una parola. Anche se non è altrettanto ovvio, <ESC>v porta invece il cursore alla pagina precedente.

Ora che si conoscono i possibili movimenti, passiamo ai comandi per manipolare il testo. Il più semplice è ^D, tramite il quale si può cancellare (Delete) il carattere su cui si trova il cursore (<ESC>d cancella la parola). Alcuni comandi di cancellazione servono anche per spostare sezioni di testo. Molti di essi, infatti, immagazzinano i blocchi di testo cancellati nei cosiddetti "kill buffer", che sono in realtà aree di sosta temporanea in memoria. Nel caso che il comando di cancellazione venga ripetuto, *Memacs* aggiunge l'ultimo brano cancellato in coda a quello già contenuto nel kill buffer.

Vediamo come effettuare un semplice spostamento: con ^K si cancella la riga dal punto in cui si trova il cursore fino alla fine; premendo di nuovo ^K si cancella anche il carattere di Ritorno Carrello al termine della riga. Per cancellare completamente una riga basta collocare il cursore all'inizio e premere due volte ^K. Impartendo il comando più volte si può cancellare un'ampia area di testo.

Ora il testo si trova nel kill buffer, e per spostarlo in una nuova posizione basta muovere il cursore nel punto desiderato e premere ^Y per "estrarre" (Yank) il contenuto del buffer. Ripetendo l'operazione, si possono introdurre nel file copie multiple di una certa parte di testo. Nel frattempo, però, si deve stare attenti a non utilizzare nessun altro comando di cancellazione, perché una volta che il contenuto sia stato "tirato fuori" (yank), ogni altro brano di testo cancellato andrebbe a sostituire quello vecchio (e non più ad aggiungersi in coda). I comandi Kill e Yank sono simili ai comandi "cut" e "paste" di molti programmi commerciali.

Abbiamo elencato otto comandi, ce ne mancano ancora due. Abbiamo imparato a spostarci nel testo e a manipolarlo; ora è necessario un modo per "salvare" i cambiamenti effettuati e un modo per uscire dal programma. I relativi comandi sono combinazioni doppie che iniziano con ^X (le sequenze doppie sono necessarie perché le sequenze singole Control e Meta non riescono a coprire tutti i comandi di *Memacs*).

Per salvare il file si digita ^X^S. Quando il lavoro è stato salvato su disco, la riga-modo vi dice quante righe di testo sono state scritte nel file dal programma. Si badi che il comando Save File pretende che l'utente, al momento di avviare

I dieci comandi base di Memacs

Sequenza	Effetto
^B	Sposta il cursore indietro di un carattere (Backward).
^X^C o ^C	Esce da Memacs (Ciao).
^D	Cancella il carattere successivo (Delete).
^F	Sposta il cursore avanti di un carattere (Forward).
^K	Cancella il testo dal punto in cui si trova il cursore fino al termine della riga (Kill).
^N	Sposta il cursore alla riga successiva (Next Line).
^P	Sposta il cursore alla riga precedente (Previous Line).
^V	Visualizza la pagina successiva (View)
^X^S	Salva il file (Save).
^Y	Estrae il contenuto del kill buffer (Y).

Altri comandi utili di Memacs

Sequenza	Effetto
^@	Imposta l'inizio di un blocco.
^R	Cerca una stringa all'indietro (Reverse search).
^S	Cerca una stringa in avanti (Search forward).
^T	Scambia due caratteri tra loro (Transpose).
^W	Cancella un blocco (Wipe off).
^X^W	Salva un file dandogli un nome.
<ESC>f	Muove il cursore avanti di una parola (Forward).
<ESC>b	Muove il cursore indietro di una parola (Backward).
<ESC>v	Visualizza la pagina precedente (View).
<ESC>u	Rende maiuscola una parola.
<ESC>l	Rende minuscola una parola.

Memacs, abbia fornito un nome di file. Se questo nome è stato omesso, o se *Memacs* è stato avviato con il doppio click del mouse sull'icona, il comando non ha successo, perché non sa in quale file deve salvare il lavoro. L'alternativa è usare il comando Save As File (^X^W) che fa comparire un prompt grazie al quale potete assegnare un nome al file in cui verrà salvato il vostro lavoro.

Per uscire definitivamente da *Memacs*, si preme ^X^C (per ricordarsene, si può pensare che uscire è come dire "ciao!" a *Memacs*). Si può anche uscire con il comando standard del CLI per bloccare una routine, ovvero ^C.

Un passo avanti

I dieci comandi elencati (e i paralleli comandi meta) sono la base essenziale per operare con *Memacs*, ma conoscerne alcuni altri semplifica molto il lavoro. Per esempio, invece di cancellare un lungo brano di testo utilizzando ripetutamente ^K, si può portare il cursore all'inizio dell'area da cancellare e impostare un "inizio blocco" digitando la combinazione Ctrl/Shift/2 (oppure ^@). Poi ci si porta alla fine dell'area e si digita ^W per cancellare (wipe off) la regione tra l'inizio blocco e il cursore: il blocco di testo viene così immagazzinato nel kill buffer.

Spesso è necessario cercare nel testo una frase o una parola specifica, e *Memacs* mette a disposizione un paio di semplici comandi di ricerca. Il principale è ^S (search), che fornisce un prompt nel quale scrivere il nome della stringa da cercare. Per dare il via alla ricerca si preme Return, e il programma "legge" il testo partendo dal punto in cui si trova il cursore e procedendo in avanti.

Se si vuole cercare la seconda ripetizione della stessa stringa si preme di nuovo ^S, ma ora *Memacs* considera la stringa precedente come quella di default, quindi basta premere Return senza riscriverla. Il comando complementare è ^R (reverse search) che cerca la stringa "all'indietro", ovvero dal punto in cui si trova il cursore fino all'inizio del testo.

Questi sono i soli comandi veramente basilari per iniziare, ma *Memacs* ne offre moltissimi altri: ce ne sono alcuni che permettono la correzione contemporanea di diversi file, e lo spostamento di brani di testo dall'uno all'altro. Ci sono comandi praticamente per qualsiasi cosa: da complesse funzioni che servono per definire le macro di comandi a cosettine semplici come trasformare le lettere di una parola in maiuscolo (<ESC>u) o in minuscolo (<ESC>l).

Iniziate con i comandi semplici. Quando vi sarete abituati, potrete esplorare i menu e la documentazione di *Memacs*: il programma vi ricompenserà con una ricca messe di efficaci funzioni. Scoprirete persino quanto può essere utile il comando che permette di scambiare tra loro i caratteri (^T). ■

MILIARDI DI BUFFER

Dovete scambiare dati tra due file di testo? C'è *Memacs*! Grazie ai suoi comandi per gestire buffer e finestre, si possono avere sott'occhio contemporaneamente due o più file.

In *Memacs*, le finestre sono aree rettangolari di larghezza pari a quella dello schermo e di altezza qualsiasi; la finestra in cui ci si trova viene definita finestra di lavoro.

Ogni finestra visualizza un buffer che contiene il file di testo su cui si sta lavorando. Ogni buffer ha un nome (in genere, le prime 15 lettere del nome del file in esso contenuto). Il primo buffer che si apre viene sempre chiamato "main". Dal momento che si possono aprire diverse finestre contemporaneamente, vi possono essere molti buffer (ovvero molti testi diversi) presenti contemporaneamente sullo schermo. L'essenziale per l'utente è capire come interagiscono tra loro i buffer e le finestre. *Memacs* non impone una relazione 1:1 tra buffer e finestre. Si può avere lo stesso buffer in diverse finestre, oppure avere buffer in memoria che non sono visualizzati in nessuna finestra.

Vi sono molti i comandi riservati a queste operazioni buffer-finestre. Con il comando Split-Window (^X2) si divide in due la finestra di lavoro, fino a una grandezza minima di due righe (se si prova a eseguire lo split di una finestra troppo piccola si riceve un messaggio di errore). Dopo aver diviso la finestra a metà, entrambe le finestre risultanti contengono inizialmente lo stesso buffer di quella originale. Si può caricare un nuovo buffer nella finestra di lavoro tramite il comando Visit (^X^V), che legge un file; il buffer originale rimane visualizzato nell'altra finestra.

Quando si sostituisce il contenuto della finestra con un nuovo buffer, quello vecchio resta in memoria anche se non è visualizzato in nessun'altra finestra. Per vedere quali sono i buffer conservati in memoria, si utilizza ^X^B: la finestra visualizza la lista dei buffer, la loro grandezza, e i nomi dei relativi file. Per sbarazzarsi della lista basta azionare il comando One Window (^X1), che ingrandisce a tutto schermo la finestra di lavoro (che non è mai quella dove compare la lista di buffer). Si può trasportare ogni buffer nella finestra di lavoro con il comando Select-buffer (^Xb) che richiede il nome del buffer da caricare.

Per rendere un po' più completo tutto questo discorso, proviamo ad agire su due file contemporaneamente. Caricate *Memacs* e una copia del file Shell-Startup, poi dividete in due lo schermo con ^X2: entrambe le finestre ora contengono un buffer di nome Shell-Startup, e quella più in alto è la finestra di lavoro. Per caricare in quest'ultima un nuovo buffer si digita ^X^V. Quando il prompt vi chiede il nome di un file, digitate Startup-Sequence e premete Return; la finestra visualizza allora la vostra sequenza di startup in un buffer di nome Startup-Sequence. Per muovervi tra le due finestre si usa ^Xn (next-window) e ^Zp (prev-window) e grazie ai comandi Kill e Yank si possono spostare brani di testo da una finestra all'altra.

METTIAMO IN RIGA GLI SPRITE

"Nick" oltre che "intaccatura" significa "momento critico", ovvero il momento in cui interviene questo utile programmino

di Mark Jordan

Una delle glorie del C-128 è la sua eccezionale famiglia di comandi per gli sprite... tutti eccetto il comando Bump. È una funzione che promette molto e mantiene poco: in teoria dovrebbe informarvi su quali sprite sono in collisione tra loro, ma l'informazione viene fornita tramite un numero che per essere comprensibile dev'essere interpretato.

Peggio ancora, Bump è così lenta che uno sprite fa in tempo a penetrare nell'altro prima che la funzione registri la collisione. E se due sprite si danneggiano a vicenda, Bump non può farci niente.

È un problema così seccante, che alla fine mi sono deciso a scrivere una mia versione di Bump, che fosse in grado d'intercettare ogni tipo di "nick", "nib" o "foul tip", ovvero ogni tipo di contatto tra gli sprite. Il risultato è stato NICK (Newly Invented Construction Kit).

Nick comunica i numeri dei primi tre sprite che entrano in collisione in un istante dato. Opera in linguaggio macchina, quindi se si vuole usare *Nick* per la programmazione bisogna installarlo e chiamarlo.

Al momento del run, il programma vi mostra otto sprite in movimento e vi tiene continuamente informati sulle collisioni dei tre più bassi (in ordine di priorità).

Per installare *Nick* nei vostri programmi, bastano la riga 10 e le righe da 60000 in poi. Queste righe installano due routine in linguaggio macchina, una delle quali relativa a un interrupt (in 2816).

Per utilizzare *Nick*, dovete per prima cosa far partire l'interrupt, e l'operazione viene compiuta dall'istruzione SYS 2816 al termine della riga 10. Poi, ogni volta che nel vostro programma avreste usato Bump (cioè ogni volta che vorreste sapere quali sprite entrano in collisione) basta che digitiate:

```
SYS 2877:RREG A,X,Y
```

SYS 2877 fa saltare il programma a una routine in linguaggio macchina, poi RREG A,X,Y ritorna dalla routine restituendo nelle variabili A, X, Y i valori dei tre sprite in collisione più "bassi" (con il numero più basso).

Se tutti gli otto sprite stanno collidendo, le variabili A, X e Y contengono i numeri 1, 2 e 3; se stanno collidendo soltanto gli sprite con numero dispari, contengono 1, 3 e 5, e così via.

Nick non è perfetto. A causa dell'intrinseca natura dei comandi sprite (basati sugli interrupt), c'è spesso un pixel di ritardo (o anche due) nella registrazione delle collisioni. Inoltre il programma riporta soltanto le prime tre collisioni. Ma a paragone del modo in cui opera Bump, queste limitazioni sono soltanto un piccolo fastidio.

A mio avviso, Bump è il pargolo deforme della nidiata del Basic 7.0. L'istinto mi suggerirebbe di sopprimerlo, ma sono un vecchio sentimentale e quindi vi fornisco *Nick*. Al resto pensateci voi.

Istruzioni: digitare il listato, salvarlo su disco e impartire il run.
Tutti i caratteri grafici e di controllo sono stati tradotti in combinazioni di tasti leggibili. Sono le istruzioni tra parentesi graffe.
Per ulteriori informazioni vedere "Come digitare i listati" a pagina 126.

```
1 REM LA RIGA 10 DEV'ESSERE ALL'INIZ
IO DEL VOSTRO PROGRAMMA
3 REM SYS 2877,0,0,0 PER LEGGERE LE
COLLISIONI SPRITE-SPRITE
4 REM SYS 2877,0,1,0 PER LEGGERE LE
COLLISIONI SPRITE-SFONDO
5 REM DOPO LE SYS USARE RREG A,X,Y P
ER TROVARE I PRIMI 3 SPRITE CHE SI S
ONO SCONTRATI. VEDERE RIGA 110
6 IF RGR(.)=5THEN PRINT"{CTRL O}IMPOS
TA IL MONITOR SUL MODO 40-COLONNE":P
RINTCHR$(27)"X":SLOW
10 FORT=0TO114:READA$:POKE2816+T,DEC
(A$):NEXT:SYS 2816
20 :
30 REM INIZIA IL PROGRAMMA DIMOSTRAT
IVO
40 COLOR0,12:GRAPHIC1,1
50 CHAR,0,0,"{3 SPAZI}",1:CHAR,0,1,"{3
 SPAZI}",1:CHAR,0,2,"{3 SPAZI}",1
60 FORT=1TO8:CHAR,0,1,STR$(T),1:SSHA
PE A$,0,2,23,22:SPRSAV A$,T:NEXT
70 FORT=1TO8:SPRITET,1,T:MOVSPRT,T*4
0,150:MOVSPRT,T*75#2:NEXT
80 GRAPHIC0,1:COLOR0,12:COLOR4,12
90 :
100 DO
110 SYS2877,0,0,0:RREG A,X,Y
120 IFA THEN CHAR,17,12,STR$(A)+STR$
(X)+STR$(Y)
130 LOOP
140 :
60000 DATA 78,A9,0D,8D,14,03,A9,0B
60010 DATA 8D,15,03,58,60,AD,19,D0
60020 DATA 29,04,F0,26,AD,1A,D0,29
60030 DATA 04,F0,17,AD,1A,D0,29,FB
60040 DATA 8D,1A,D0,AD,1E,D0,8D,F0
60050 DATA 0B,AD,1F,D0,8D,F1,0B,4C
60060 DATA 33,FF,AD,1A,D0,09,04,8D
60070 DATA 1A,D0,4C,65,FA,BD,F0,0B
60080 DATA A2,01,A0,00,8C,F2,0B,8C
60090 DATA F3,0B,8C,F4,0B,18,4A,90
60100 DATA 0B,48,8A,99,F2,0B,68,C8
60110 DATA C0,03,F0,05,E8,E0,09,D0
60120 DATA EC,A9,00,8D,F0,0B,8D,F1
60130 DATA 0B,AD,F2,0B,AE,F3,0B,AC
60140 DATA F4,0B,60
```

■

GLI OFFSET ALLE STRUTTURE PER L'AMIGABASIC

Un approccio "pratico" all'AmigaBASIC, con alcuni esempi che mettono in una nuova luce le possibilità di questo linguaggio

di Bryan D. Catley

L'AmigaBASIC è una potente versione di Basic, ma ha un limite: non permette un accesso diretto a molte strutture del ROM Kernel. Se però ci equipaggiamo con le funzioni WINDOW(7) e WINDOW(8), con i comandi PEEK, PEEKW, PEEKL, POKE, POKEW e POKEL, con alcune chiamate al sistema operativo e alcuni offset alle strutture, possiamo riuscire a estrarre alcune routine utili ed efficaci.

Anziché farvi un mucchio di discorsi teorici scarsamente applicabili in pratica, vi mostrerò come ottenere alcuni risultati che non avreste mai pensato di poter raggiungere con l'AmigaBASIC.

Le funzioni WINDOW(7) e WINDOW(8) sono gli strumenti di base per questo lavoro di scavo, dal momento che forniscono i due punti di partenza necessari per accedere alle varie strutture. WINDOW(7) procura l'indirizzo (o un puntatore all'indirizzo) della struttura Window della finestra attiva, e WINDOW(8) procura l'indirizzo della struttura Rastport della finestra attiva (che ne definisce i parametri di visualizzazione). Quanto ai tool, PEEK e POKE accedono a singoli byte; PEEKW e POKEW accedono a una word (due byte); PEEKL e POKEL accedono a una long word (quattro byte).

Come recuperare un valore nella palette

Il comando PALETTE dell'AmigaBASIC permette d'impostare un registro di colore con un valore qualsiasi. Il guaio è che l'AmigaBASIC non fornisce alcun metodo diretto per scoprire qual è il colore in uso. I seguenti sotto-programmi aggirano questa restrizione, restituendo i valori di rosso, verde e blu (valori RGB) per la palette indicata. Si noti che mentre l'AmigaBASIC usa soltanto i valori 0 e 1, il sistema operativo usa valori da 1 a 15, ed è questo il motivo per cui viene eseguita una divisione per 16.

```
SUB GetRGB (pal%, r, g, b) STATIC
'ottiene il valore di ViewPortAddress
'per la finestra
vpa&=ViewPortAddress&(WINDOW(7))
'ottiene il puntatore alla tavola dei colori
colorTbl&=PEEKL(PEEKL(vpa&+4)+4)
'calcola i valori Basic della palette
r=PEEK(colorTbl&+2*pal%)/16
bluegreen=PEEK(colorTbl&+2*pal%+1)
g=(bluegreen/16)/16
b=(bluegreen MOD 16)/16
END SUB
```

Il seguente segmento di codice dà una dimostrazione d'uso del sotto-programma.

```
LIBRARY "intuition.library"
DECLARE FUNCTION ViewPortAddress&()
LIBRARY
FOR p%=0 TO WINDOW(6)
  CALL GetRGB(p%, red, green, blue)
  PRINT "Palette=";p%;red;green;blue
NEXT
LIBRARY CLOSE
END
```

All'inseguimento del mouse

Per tenere sotto controllo i movimenti del mouse si possono utilizzare, al posto delle apposite funzioni, i dati contenuti nella struttura Window. Dal momento che si ricorre a PEEK per ottenere i valori direttamente, questo metodo è più rapido di quello convenzionale. L'esempio seguente applica questa tecnica a un semplicissimo programma di disegno: si muove il mouse per disegnare, si preme un tasto per uscire.

```
PRINT "Premi un tasto per uscire"
'continua finche' non viene premuto un tasto
WHILE INKEY$=""
  'calcola la coordinata y del mouse
  yMouse%=PEEKW(WINDOW(7)+12)-11
  'calcola la coordinata x del mouse
  xMouse%=PEEKW(WINDOW(7)+14)-4
  'riempie i pixel
  PSET (xMouse%, yMouse%)
WEND
END
```

Si notino le necessarie correzioni per il titolo della finestra e per il bordo. Se non fossero state incluse, il punto di disegno sarebbe stato separato dal mouse.

Un Workbench a due colori

Se avete bisogno di memoria, potete risparmiare 16K impiegando uno schermo Workbench a due colori al posto di quello solito a quattro colori. Il seguente programma serve appunto per dare allo schermo questa impostazione. Attenzione! Il solo modo per tornare al normale Workbench è rifare il boot della macchina.

```
LIBRARY "intuition.library"
LIBRARY "graphics.library"
'ottiene un puntatore alla struttura BitMap
maps&=PEEKL(WINDOW(8)+4)
'ottiene il numero di bitplane
numPlanes%=PEEK(maps&+5)
```

```
'ottiene il puntatore alla struttura Screen
scrnStruct&=PEEKL(WINDOW(7)+46)
'ottiene altezza e larghezza dello schermo
sWd%=PEEKW(scrnStruct&+12)
sHd%=PEEKW(scrnStruct&+14)
'imposta un nuovo numero di bitplane
POKE maps&+5,1
'ci si sbarazza dei bitplane in eccesso
FOR n%=numPlanes% TO 2 STEP -1
planePtr&=PEEKL(maps&+4+4*n%)
CALL FreeRaster(planePtr&, sWd%, sHt%)
CALL RemakeDisplay
CALL RefreshWindowFrame(WINDOW(7))
CLS
NEXT
LIBRARY CLOSE
SYSTEM
```

Uno schermo senza bordi

L'AmigaBASIC non fornisce alcun metodo diretto per eliminare i bordi di una finestra, ma un sistema c'è: basta modificare le appropriate strutture e ridisegnare il contorno della finestra, come fa la seguente routine. Si osservi che, anche se il bordo scompare, non si può comunque avere accesso a quella zona di schermo.

```
SUB RemoveBorder STATIC
'ottiene un puntatore ai modi window
winModes&=Window(7)+24
'ottiene il modo
mode&=PEEKL(winModes&)
'imposta il modo niente-bordo
mode&=mode& AND (2^ 26-1-1024)
mode&=mode& OR 2048
POKEL winModes&, mode&
'ridisegna la cornice (cancellando il bordo)
CALL RefreshWindowFrame(WINDOW(7))
END SUB
```

Ed ecco un modo per sfruttare il precedente sottoprogramma:

```
LIBRARY "intuition.library"
CALL RemoveBorder
LIBRARY CLOSE
END
```

Ricolorare il bordo della finestra

Se si intende mantenere il bordo della finestra, ma si vuole cambiarne il colore, si possono utilizzare i seguenti codici.

```
SUB ChangeBorderColor(newDetail%,newBlock%) STATIC
'ottiene i puntatori alle penne in uso
detailPen&=WINDOW(7)+98
blockPen&=WINDOW(7)+99
'imposta le nuove penne
POKE detailPen&, newDetail%
POKE blockPen&, newBlock%
'ricolora la cornice della finestra
CALL RefreshWindowFrame(WINDOW(7))
END SUB
```

Si può mettere alla prova la routine ricorrendo a questo segmento di programma:

```
LIBRARY "intuition.library"
FOR block%=0 TO 3
  FOR detail%=0 TO 3
    ChangeBorderColor detail%,block%
    FOR n=1 TO 1000: NEXT
  NEXT
NEXT
ChangeBorderColor 0,1
LIBRARY CLOSE
END
```

Limitare la grandezza di una finestra

Con POKEW si possono limitare le dimensioni massime dell'ingrandimento di una finestra (offset 8 contiene la larghezza della finestra e offset 10 l'altezza. Per accedere a quei valori si usi PEEKW). Per limitare la grandezza massima seguite i passi che ora indicheremo, ma tenete a mente una cosa: la grandezza minima deve *sempre* essere inferiore a quella massima, e la grandezza massima deve *sempre* essere superiore a quella della finestra in uso.

```
winStruct&=WINDOW(7)
POKEW winStruct&+16, newMinimumX%
POKEW winStruct&+18, newMinimumY%
POKEW winStruct&+20, newMaximumX%
POKEW winStruct&+22, newMaximumY%
```

Più grande e più nero

Quando si seleziona l'opzione "bold" (nero) per la stampa di un testo, i caratteri vengono stampati due volte: nella posizione normale e un pixel più a destra. È possibile creare un testo ancora più nero, accrescendo la separazione. Per esempio:

```
'ottiene il puntatore alla struttura Font in uso
currentFont=PEEKL(WINDOW(8)+52)
'attiva il nero
POKE WINDOW(8)+ 56,2
'imposta il nero al nuovo offset 4
POKEW currentFont&+28,4
PRINT "Come ti sembra, cosi'?"
'torna alla condizione normale
POKE WINDOW(8)+56,0
```

Si tenga a mente che Print stampa un carattere alla volta, con il risultato che stili come il nero, il corsivo e così via hanno una parte di ciascun carattere troncata dal successivo e l'effetto complessivo non è pienamente visibile. Come alternativa, la funzione di disegno "text" costruisce un'immagine grafica della stringa caratteri e quindi la disegna sullo schermo nella locazione appropriata.

C'è ancora qualcosa da scoprire?

Con questi esempi abbiamo appena scalfito la superficie di quello che potremmo fare. Si potrebbero per esempio sfruttare i gadget di Intuition, impostare sotto-menu e moltissime altre cose. Per fare uso di queste opzioni, in linea di massima le operazioni da compiere sono: un'allocazione di memoria, la costruzione delle strutture da alterare con POKE, POKEW e POKEL, il collegamento di queste strutture con le strutture Window o Rastport. Come avete visto, comunque, dovete usare gli offset e non i nomi simbolici... preparatevi a fare qualche calcolo.

■

COMINCIAMO DA ZERO: PARTE PRIMA

Iniziamo una breve serie di articoli per aiutare i nuovi utenti dell'Amiga a fare i primi passi con il sistema operativo del loro computer

di Mark L. Van Name e Bill Catchings

Una delle principali difficoltà nel tenere una rubrica regolare sull'AmigaDOS è che praticamente tutti gli utenti hanno a che fare con il sistema operativo, sia i novellini che i superesperti. E allora, a quale livello si devono trattare gli argomenti? Abbiamo deciso di cominciare dedicandoci ai nuovi utenti, parlando delle caratteristiche fondamentali del Workbench *e del* CLI. *Facciamo le nostre scuse a chi ha già superato questa fase, e promettiamo che in futuro penseremo anche a lui.*

La prima cosa che dovete fare con il vostro nuovo disco *Workbench*, è la copia di backup (termine con il quale si indicano le copie di riserva). In effetti è la prima operazione da effettuare con qualunque programma si acquisti. Una volta fatta la copia, consigliamo di riporre l'originale in un posto sicuro e di utilizzare come "*Workbench* di lavoro" la copia. L'originale, in realtà, dovrebbe essere utilizzato soltanto per fare nuove copie di backup.

Per effettuare la copia, oltre all'originale bisogna avere a disposizione un disco vuoto. Ogni disco da 3,5" ha in alto a destra, dalla parte dell'etichetta, una finestrella quadrata. Sul retro della finestrella c'è una linguetta di plastica scorrevole nota come "protezione". Quando il quadratino copre la finestrella, il disco è "write enabled", ovvero abilitato alla scrittura: l'Amiga può "scrivere" dati sul disco. Quando invece il quadratino è in posizione retratta (ossia appare la finestrella), il disco è "write inhibited" o "write protected", ovvero protetto dalla scrittura: l'Amiga non può scrivervi nessun dato. Proteggete il disco *Workbench* originale in modo che non possa essere alterato in caso di errore, e assicuratevi invece che sul disco vuoto si possa "scrivere".

In questo breve articolo, faremo l'ipotesi che abbiate un Amiga standard, con un solo disk drive. Ora eseguite il "boot" (l'avvio) della macchina, inserendo nel disk drive il disco *Workbench* originale.

I primi passi

Quando appare lo schermo Workbench, vedrete nell'angolo superiore destro dello schermo una piccola "icona-disco" etichettata con la scritta "Workbench 1.3". Portate il puntatore del mouse su quell'icona e premete una volta il pulsante sinistro. L'icona diventerà bianca: l'avete "selezionata".

Adesso muovete il puntatore del mouse sulla barra bianca dei menu nella parte alta dello schermo. Quando il puntatore si trova sulla barra, premete il pulsante destro del mouse e tenetelo premuto. Appariranno tre nomi di menu: Workbench, Disk e Special.

Sempre tenendo premuto il pulsante destro del mouse, muovete lentamente il puntatore su ognuna delle tre parole; non appena il puntatore le tocca, le parole vengono evidenziate e sotto compare un elenco di "opzioni" (le possibili scelte che offre il menu). Selezionate l'opzione "Duplicate" dal menu Workbench.

In alto a sinistra sullo schermo apparirà un rettangolo, che vi chiederà di:

```
Put Workbench 1.3
( FROM disk ) in drive DF0:
```

ovvero, di "inserire il disco Workbench nel drive DF0:".

I rettangoli come questo vengono chiamati "requester". Nel requester, sotto alle istruzioni, ci sono due parole: *Continue* (continua) e *Cancel* (cancella), che rappresentano i cosiddetti "gadget di uscita". Per rispondere al requester, premete il pulsante sinistro del mouse su uno dei due gadget (generalmente i gadget svolgono compiti piuttosto ovvi; nel nostro caso, Continue fa continuare l'operazione che stiamo svolgendo, mentre Cancel la interrompe).

L'altro strano termine contenuto nel requester, ovvero "drive DF0:" è il nome assegnato dall'Amiga al disk drive interno (il nome "DF1:" viene invece assegnato al primo disk drive esterno per l'A500 o per l'A1000, e al secondo disk drive interno per l'A2000. In caso di ulteriori drive esterni si adottano i nomi DF2:, DF3: e così via). Dal momento che avete appena eseguito il boot dell'Amiga, il disco Workbench si trova già nel disk drive DF0:. Di conseguenza premete pure il pulsante sinistro del mouse su Continue.

Leggere e scrivere

Ora l'Amiga sta immagazzinando in memoria una parte dei dati contenuti nel disco *Workbench*. Durante la lettura, il requester viene sostituito da un "messaggio di stato" del tipo:

```
reading nn, mm to go
```

ovvero "letti nn, mm da leggere". Quelli che abbiamo chiamato "nn" e "mm" sono in realtà numeri interi compresi tra 0 e 79, che rappresentano le ottanta "aree di memoria" principali (dette anche "cilindri") di un disco dell'Amiga.

Dopo qualche istante, il messaggio di stato scompare, e al suo posto arriva un altro requester che dà una nuova istruzione, chiedendovi di:

```
Put DESTINATION disk
```

```
( TO disk) in drive DF0:
```

ovvero di "inserire il disco destinazione nel disk drive DF0:". Togliete quindi il disco *Workbench* e mettete al suo posto il disco vuoto. Dopo la sostituzione, premete di nuovo il pulsante del mouse su Continue, e avrà inizio la seconda parte del procedimento, quella di scrittura. Sullo schermo comparirà il messaggio:

```
writing nn, mm to go
```

ovvero, "scritti nn, mm da scrivere". L'intero procedimento si ripeterà un paio di volte. Ogni volta dovrete inserire il disco *Workbench*, aspettare che l'Amiga ne legga una parte, poi inserire il disco vuoto, attendere che vi vengano scritti i dati memorizzati, e così via. Questi cambi di disco sono dovuti al fatto che il comando di copia dell'Amiga non può immagazzinare in memoria il contenuto del disco *Workbench* tutto in una volta, e la stessa cosa capiterà anche con altri programmi.

A proposito! Non dovete mai togliere un disco dal disk drive finché la spia rossa non si è spenta, altrimenti potreste danneggiarlo.

Durante il procedimento di copia, al disco vuoto viene assegnata un'icona personale, contrassegnata dalla scritta "DF0:busy" (DF0:occupato). Quando la copia è finita, il requester scomparirà e all'icona del disco vuoto verrà assegnato un nuovo nome: "copy of Workbench 1.3" (copia del *Workbench 1.3*). Mettete via l'originale e rieffettuate il boot del computer con la copia (per rieffettuare il boot è sufficiente premere contemporaneamente il tasto Control e i due tasti Amiga, a sinistra e a destra della barra spaziatrice).

Cambiare il nome a un disco

Prima di attaccare al disco un'etichetta (come si dovrebbe fare con tutti i dischi, per sapere immediatamente qual è il loro contenuto) completeremo la lezione sul procedimento di copia aggiungendo un piccolo dettaglio: il cambiamento del nome. Nel nostro caso, il nome "copy of Workbench 1.3" è un nome troppo lungo per un disco che verrà usato quotidianamente, quindi è meglio cambiarlo. Per prima cosa, selezionate la relativa icona. Poi selezionate l'opzione Rename del menu Workbench.

Un rettangolo lungo e sottile, contenente il nome del disco, comparirà al centro dello schermo. Ora potete cambiarlo utilizzando i tasti di correzione (questo rettangolo, dal momento che permette di correggere una stringa di testo, viene chiamato "gadget stringa"). Muovetevi avanti e indietro impiegando le frecce sinistra e destra, cancellate usando i tasti Delete e Backspace, e digitate tutte le lettere e i numeri che volete. Potete per esempio abbreviare il nome in "Workbench1.3", o scegliere un nome del tutto nuovo, anche se vi suggeriamo di attenervi sempre al nome più ovvio.

Formattare un disco

Siamo quasi pronti a cominciare l'esplorazione delle utility del *Workbench*, ma c'è bisogno di un mezzo per immagazzinare i propri dati. Il disco *Workbench* è praticamente pieno, e comunque non sarebbe certo una buona idea registrare i dati proprio lì, quindi è necessario un altro disco vuoto.

Sfortunatamente, non è possibile prendere semplicemente un disco nuovo e infilarlo nel disk drive. Prima si deve "formattarlo". Assicuratevi sempre di formattare dischi vuoti oppure contenenti dati che non vi servono più, perché la formattazione distrugge tutti i dati contenuti nel disco.

Togliete dunque dal disk drive il disco *Workbench* e inserite quello vuoto. Apparirà sullo schermo un'icona con la scritta "DF0:BAD"; selezionatela, e poi selezionate Initialize dal menu Disk. A questo punto un requester vi chiederà di rimettere nel disk drive il disco *Workbench*. Compiuta l'operazione, comparirà un altro requester che vi chiederà di sostituirlo di nuovo con il disco vuoto (che viene chiamato "disco destinazione"). Il processo di scambio di dischi andrà ripetuto ancora una volta. (Stanchi d'inserire dischi? Ora vi rendete conto perché ci sono tanti utenti dell'Amiga che acquistano un hard disk oppure un disk drive esterno. È una soluzione che consigliamo a tutti).

Al termine degli scambi comparirà come sempre un requester, che vi chiederà:

```
Ok to Initialize disk
in drive DF0:
(all data wil be erased)?
```

ovvero: "Ok per l'inizializzazione del disco nel drive DF0: (tutti i dati verranno cancellati)?"

Se invece il messaggio visualizzato nel requester fosse stato:

```
Ok to Initialize volume
[name]
(all data will be erased)?
```

ovvero: "Ok per l'inizializzazione del volume, [nome], (tutti i dati verranno cancellati)?" avrebbe voluto dire che il disco non era un disco nuovo e non formattato, ma un disco già formattato in precedenza che probabilmente contiene dati che non volete perdere ("volume" è uno stravagante sinonimo di "disco").

È la vostra ultima possibilità di fare marcia indietro se avete accidentalmente inserito un disco contenente dati che non devono essere cancellati.

Quando siete pronti, premete il pulsante del mouse su Continue. Durante la formattazione apparirà un altro messaggio:

```
Formatting cyl nn, mm to go
Verifying cyl nn, mm to go
```

ovvero "Sto formattando il cilindro nn, ne mancano mm; sto verificando il cilindro nn, ne mancano mm", dove nn e mm sono gli stessi numeri di cilindro che abbiamo già incontrato nel caso del processo di copia.

Quando la formattazione è finita, apparirà un breve messaggio:

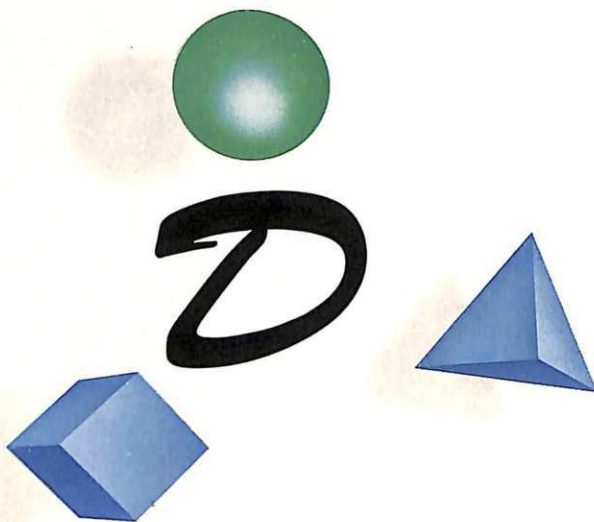
```
Initializing disk
```

ovvero "Sto inizializzando il disco".

Compiuta anche questa operazione, siete pronti a cominciare. Il nome che compare ora sotto l'icona del disco è "empty" (vuoto), un nome non molto espressivo. Conviene quindi cambiarlo, ricorrendo alla tecnica già illustrata, e assegnargli il nome più adatto all'uso che avete in mente.

Nel prossimo articolo spiegheremo come il *Workbench* "vede" i file di dati, e come si fa a caricare e memorizzare i file su un disco nuovo e formattato. Intanto fate qualche esperimento con il vostro nuovo Amiga! ■

COMMODORE E DIDATTICA

ESPERIENZE DIDATTICHE A CONFRONTO

LA PAURA DI EVOLVERSI

Grazie alla collaborazione con il professor Franzato, di Mestre, Commodore Gazette affronterà nei prossimi numeri il difficile argomento dell'informatica nelle scuole. Questo articolo introduttivo mette a fuoco il problema e sarà seguito da alcune proposte di soluzioni.

Chi per lavoro trascorre parte delle proprie giornate tra i banchi di scuola non può negarlo: sa che, al di là del gran parlare di computer, software didattico, piano ministeriale per la formazione dei docenti o altre iniziative del genere, la quotidiana realtà scolastica è ben diversa. La maggioranza dei docenti di qualsiasi ordine e grado non è affatto in grado di utilizzare un elaboratore per insegnare meglio la propria materia. Ci rendiamo conto che l'affermazione è piuttosto dura e categorica, ma riteniamo che smentirla non sia facile.

In quante scuole il docente di Lettere insegna a creare un testo – a fare "il tema" – col word processor? Quale insegnante di Liceo artistico o Accademia di Belle Arti parla con i suoi studenti dell'argomento computer-art o computergrafica?

C'è molta curiosità, è vero. Ma una volta rientrati nelle aule, riprende il solito tran tran. E allora viene da chiedersi: perché? Perché esiste questa situazione? Che cosa rende tanto difficile agli insegnanti accostarsi al computer? Quali sono le ragioni e i meccanismi psicologici della loro riluttanza a usarlo nel modo didatticamente più appropriato? Quali le ragioni che li spingono a rifiutarlo a priori, evidenziando quanto più possibile le sue limitazioni in campo psicopedagogico? Che sia soltanto pigrizia? È vero che prima di saper usare con sicurezza un personal ci vuole tempo... e il tempo e la voglia sono difficili da trovare. Ma forse c'è dell'altro. Il computer suscita indefinibili paure indistinte a cui non sappiamo o non vogliamo dare un nome.

Questa rubrica si basa su testimonianze di alunni ed insegnanti, che si interessano alla didattica del computer.
I contributi editoriali sono grandemente aprezzati.
Inviate eventuali materiali (articoli, foto, disegni, descrizioni di esperienze...) a:

COMMODORE GAZETTE
Comodore e didattica
Via Monte Napoleone, 9
20121 Milano

Ignoto

I motivi possono essere molti. Cerchiamo di prenderne in esame alcuni.

Le difficoltà d'approccio possono iniziare nel momento in cui il computer è sul tavolo davanti a noi; ma anche prima, quando cioè, per le più svariate circostanze, scopriamo che dovremo avere a che fare con lui e, bene o male, conviverci. Il timore, ancora vago e imprecisato, che proviamo davanti alla tastiera e al monitor (oggetti freddi, inusitati, poco familiari), può essere ricondotto al timore ancestrale dell'ignoto; e quest'ultimo – per dirla con le parole di un maestro dei racconti dell'orrore del Novecento, H. P. Lovecraft – è "il genere di paura più antico e forte". Temiamo che, toccando quei tasti, possa accadere qualcosa d'irreparabile e dannoso, qualcosa che sfugga al nostro controllo intellettuale, personale e collettivo. A consolidare e accrescere questa paura, contribuiscono non di poco e non di rado i mass media, che danno spesso informazioni inesatte e/o affrettate, e persino le trame di alcuni film o romanzi di fantascienza. Il primo e più conosciuto esempio che viene alla mente è quell'HAL 9000 di *2001: Odissea nello spazio* (c'è persino chi insinua che dietro alle tre lettere H, A, L, si nasconda la sigla di una famosissima industria di computer: provate ad andare avanti di una lettera...).

Pensiero

Altra causa di diffidenza nei riguardi dell'elaboratore è il modo di pensare, rigido e non ambiguo, a cui il suo impiego tende ad abituarci. Si può ritenere che un simile modo di pensare – analitico ed euristico – atrofizzi il nostro spontaneo pensiero creativo, fantasioso e immaginativo: quel tipo di pensiero che gli studiosi di psicopedagogia chiamano "pensiero divergente". Qualcosa di vero c'è, perché al momento della programmazione dobbiamo in effetti esprimerci con un linguaggio simbolico dotato di una sintassi e un lessico relativamente limitati e quindi rigidi; d'altro canto, trovare la maniera più efficace per comunicare al computer dati e programmi affinché il risultato finale sia quello ideato e immaginato, è compito che spetta innegabilmente proprio al pensiero divergente. Creativo, appunto. E poi anche la lingua con cui parliamo e scriviamo tutti i giorni è fatta dopotutto di un lessico, una sintassi e una grammatica. Non per questo, a quanto pare, è stata d'ostacolo alla comunicazione spicciola, e nemmeno alla creazione d'innumerevoli capolavori. Altrimenti la letteratura non esisterebbe.

Scrive Paolo Lollini nel suo libro *Didattica e computer: quando e come il computer nella scuola* (Brescia, Editrice La Scuola, 1985): «Il computer possiede una necessità logica che lo fa rispondere agli stimoli in maniera matematicamente determinata. Ma il prodotto delle scelte, che possono essere infinite, dipende da chi lo usa e non ha nulla di deterministico, di rigido e di freddo. Non più di quanto ne abbia la logica architettura di una fuga di Bach». È comprensibile che alcuni insegnanti trovino delle difficoltà e siano poco disposti, dopo anni di consolidata esperienza, ad accettare e fare propria la mentalità informatica e la diversa visione del reale in essa implicita. Ma lo sforzo non è poi enorme né impossibile: basta provare. E soprattutto volere.

Emotività

Ma, probabilmente, la più forte crisi emotiva che l'elaboratore può provocare nasce quando si comincia a considerarlo – più o meno consciamente – un'estensione della mente, un cervello esterno che può sbilanciare il nostro io più profondo, facendoci mancare i più intimi riferimenti psicologici che sostengono la fede in noi stessi e, con questa, la nostra personalità.

Questa umanissima paura è poi rafforzata da tutto ciò che si sente dire e qua e là si legge sull'intelligenza artificiale, sui modelli computazionali della mente, per non parlare degli elaboratori neuronali e dei microprocessori costruiti su base biologica che consentirebbero la realizzazione dei cosiddetti "biocomputer": i computer della VI generazione, di cui (per fortuna) non si sa ancora molto. Simili macchine rimangono per ora confinate nei laboratori di ricerca, protette e coperte dal segreto militare. Ma anche senza pensare a questi elaboratori ai limiti della fantascienza, essere vagamente coscienti dell'esistenza di una macchina che imiterebbe i nostri processi mentali e che un giorno (non lontano) potrebbe essere capace di elaborarli con velocità, efficienza ed efficacia molto superiori alle nostre, è una visione non certo rassicurante del futuro.

Romanticismo

Eppure, per dirla con Sherry Turkle, autrice del libro *Il secondo io. Il computer e l'uomo: convivere, amarsi, capirsi* (Milano, Frassinelli, 1985) il cui soggetto sono le modifiche psicosociali provocate dall'era informatica, la reazione alla cultura dell'elaboratore è "romantica". Guardando alla mente come a un modello di "macchina multiprocessiva", costituita cioè da programmi diversi e distinti che interagiscono contemporaneamente, se ne evidenziano gli inspiegabili e ineffabili aspetti non meccanici, ossia i sentimenti. Dice David, un programmatore dodicenne: «Quando ci saranno computer bravi come persone, i computer faranno un sacco di lavori, ma ci saranno ancora cose da fare per gli uomini. Dirigeranno i ristoranti, assaggeranno i cibi e saranno quelli che si amano, hanno famiglia e si vogliono bene. Credo che saranno anche quelli che andranno in chiesa».

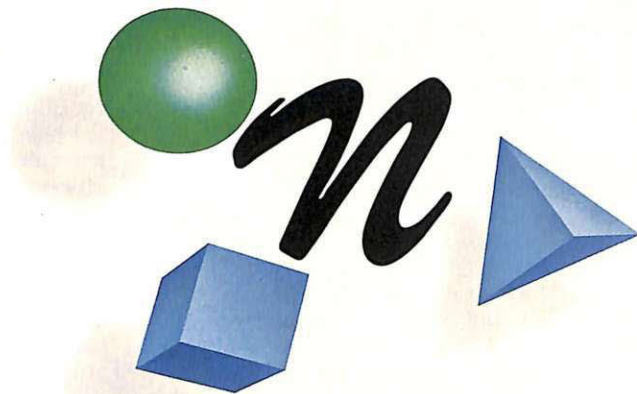
Evoluzione

Ma forse, ciò che spesso provoca un senso di fastidio e quindi di rifiuto verso l'uso didattico del computer, non è tanto un'indefinita paura quanto piuttosto l'imbarazzo che proverebbero gli insegnanti nel vedere i loro alunni maneggiare queste macchine con una sicurezza, una naturalezza e una familiarità nettamente superiori alle loro; il che li mette in crisi: per la prima volta vedrebbero messa in dubbio la loro cultura e, al contempo, ribaltato il loro status di docenti. E quest'imbarazzo di certo aumenterebbe col diminuire dell'età dei discenti.

Tutte queste motivazioni, questi timori, queste idiosincrasie sono senz'altro comprensibili. È vero: non è facile dialogare e convivere con l'elaboratore. Ma con un po' di pazienza si può imparare. A patto che ci sia la volontà di farlo, perché se questa manca in partenza, allora è lecito sospettare che tutte le paure fin qui menzionate servano sostanzialmente a nasconderne un'altra, di cui veramente dovremmo spaventarci. Un tipo di paura che ci blocca e ci impedisce, come insegnanti, di crescere e migliorare di pari passo alla crescita e alle innovazioni sociali prodotte dalla tecnologia. Un tipo di paura che riusciamo a definire soltanto come la paura di evolversi.

di Stefano Franzato

Oltre ai testi che abbiamo citato nel corso dell'articolo suggeriamo anche quello di Gianni Zanarini, L'emozione di pensare. Psicologia dell'informatica *(Milano, CLUP-CLUED, 1985). Le ricerche di Sherry Turkle (e di molti altri studiosi) sulle nuove tematiche dell'intelligenza artificiale e dei modelli computazionali della mente sono ampiamente illustrate e commentate nel recente e interessante volume dal titolo* La macchina e la mente *di George Johnson (Milano, IHT Gruppo Editoriale, 1990). Dei biochip e delle frontiere più lontane e segrete nella ricerca sull'intelligenza artificiale si parla nel volume* I creatori del domani *di Grant Fjermedal (Milano, IHT Gruppo Editoriale, 1990).*

■

ITALIA

OCCHIALI CONTRO LO STRESS VISIVO

All'avanguardia nella realizzazione di soluzioni per ridurre lo stress visivo (sperimentate già con successo per gli schermi televisivi), la Polaroid Italia ha ora lanciato sul mercato la lente TV-TECH MK II, specificamente studiata per gli operatori ai videoterminali. Color marrone-grigio, con un assorbimento medio del trenta per cento, queste lenti hanno l'effetto di aumentare il contrasto

dell'immagine rendendo migliore la sua definizione. Vengono così attenuati i tipici disturbi visivi di cui è vittima chi è costretto a lavorare per parecchie ore davanti a un monitor: al generale miglioramento della qualità dell'immagine si associa un effetto di "taglio" delle radiazioni UVA e UVB e di assorbimento globale superiore al settanta per cento della luce blu (che causa immagini sfocate, danneggiando la nitidezza del campo visivo).

La lente, scarsamente selettiva, non altera in modo evidente la cromaticità, rafforzando invece le tonalità del rosso e rendendo più scuri i toni della banda del blu; il superdiafal, uno speciale trattamento a cinque strati, elimina i riflessi delle luci di disturbo circostanti, dando ai TV-TECH MK II la caratteristica di occhiali "riposa vista". La loro versatilità li rende adatti sia a prevenire i disturbi dovuti ai difetti dei videoterminali (tremolii, scarsa definizione, mancanza di rotondità dei tratti dell'immagine "a punti") sia per ridurre l'affaticamento visivo provocato da ambienti di lavoro con luminosità intensa.

Li abbiamo provati in redazione, e i risultati sono stati più che soddisfacenti.

Divisione sunglasses
Polaroid (Italia) S.p.A.
Via Piave, 11
21051 Arcisate (VA)
(Tel. linea verde Polaroid 167820050)

GRATIS C1-TEXT 2.0

I possessori delle versioni precedenti la 2.0 di C1-TEXT possono richiedere gratuitamente la nuova versione inviando per posta il loro vecchio disco (soltanto il disco 1) e allegando una busta già affrancata per la restituzione. La scheda di registrazione, se non è già stata spedita, va compilata e allegata.

Cloanto Italia
Via G. B. Bison, 24
33100 Udine

UN NUOVO PC DALLA COMMODORE

Il PC60-III è l'ultimo nato di casa Commodore; si tratta del più potente computer mai annunciato dalla CBM, specificamente concepito per gli utenti professionali e aziendali. È dotato di microprocessore Intel 80386DX, di porte per interfacce parallele e seriali IBM compatibili e di una porta per il collegamento del mouse Microsoft compatibile; è disponibile in tre configurazioni.

Werter Mambelli, direttore generale della Commodore Italiana, esponendo i dati delle ricerche di mercato che hanno anticipato la realizzazione del nuovo prodotto, ha presentato il PC60-III come un computer molto versatile, in grado di attrarre l'attenzione di una vasta fascia di specialisti. «Poiché questo tipo di utente è un investimento per l'azienda in cui opera, il suo tempo è limitato e prezioso. Attendere che un foglio elettronico esegua i suoi calcoli, che un file venga caricato e, ancora, che un grafico venga realizzato in tutti i

suoi dettagli, può rivelarsi un'esperienza frustrante, oltre che svantaggiosa in termini economici. Il PC60-III, oltre che risolvere tutti questi problemi» ha dichiarato Mambelli «offre al contempo un'elevata capacità di espansione che consente di tutelare l'investimento iniziale». Infatti due dei nove slot di espansione sono collegati al processore tramite un bus di memoria brevettato, a 32 bit, con cache SRAM da 32 KB, che consente all'utente di utilizzare anche le applicazioni che richiedono un esteso ricorso alla memoria su disco, senza che le prestazioni del sistema vengano ridotte.

La configurazione standard prevede due disk drive (da 3,5" e 5,25") e 2 MB di memoria centrale (espandibile fino a 18 MB). Le altre due sono dotate di hard disk (80 e 200 MB espandibili fino a 400 MB).

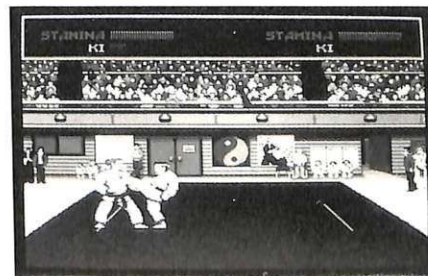
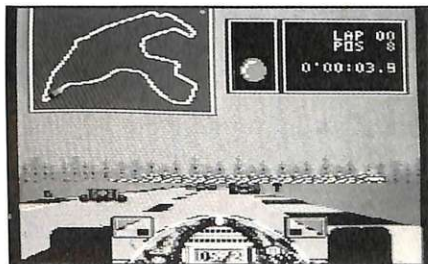
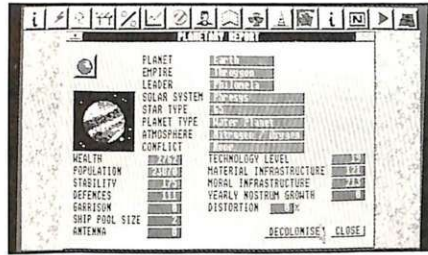
Il PC60-III è inoltre dotato del software MS-DOS 4.01, GW-Basic 3.23, tastiera internazionale a 102 tasti e monitor monocromatico VGA da 14".

Commodore Italiana spa
(PC60-III, L. 8.100.000 mod. 1,
L. 9.100.000 mod. 80,
L. 11.300.000 mod. 200).
Viale Fulvio Testi, 280
20126 Milano
(Tel. 02/661231)
(Fax 02/66101261)

ESTERO

ELECTRONIC ARTS NEWS

Durante la cena in onore delle industrie che si sono meglio distinte durante il 1989, tenutasi il 18 dicembre scorso, la Commodore Inglese ha insignito la Electronic Arts del *Premio Enterprise & Vision*. La software house inglese si è aggiudicata il premio grazie a una delle sue realizzazioni di punta, **Deluxe Paint III**, riconosciuto come il prodotto che più di ogni altro ha saputo valorizzare le caratteristiche dell'Amiga nel campo della computergrafica.

Nel corso della stessa serata sono stati poi assegnati due importanti riconoscimenti alla **Bullfrog Productions**: quello di "Programmatori dell'anno" e quello di "Miglior gioco originale dell'ultimo decennio" per *Populous*. Da più parti *Populous* viene considerato il vero mattatore dell'anno, essendosi aggiudicato (oltre al

Da sopra a sotto: i nuovi giochi prodotti dalla Electronic Arts per i prossimi mesi. Imperium, 688 Attack Sub, Ferrari Formula One e Budokan: The Martial Spirit

premio) anche un vastissimo consenso di pubblico, testimoniato dalle oltre 80 mila copie vendute (in formato ST e Amiga) in Europa. Il successo del gioco nel formato PC ha portato le vendite a toccare la soglia delle 100 mila copie, nel periodo natalizio.

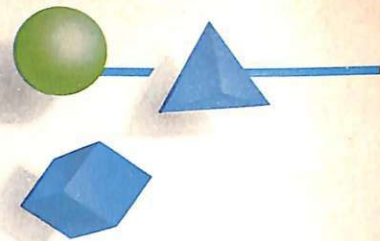
La Electronic Arts ha da poco annunciato il lancio sul mercato di **Imperium**, un nuovo gioco strategico ambientato in un complesso scenario ad altissima tecnologia. Gli autori, Matthew Stibbe e Nick Wilson, entrambi nomi noti nel campo dei videogiochi, hanno realizzato un futuro in cui ogni giocatore è chiamato a cimentarsi con l'economia, la politica, la diplomazia e il potere militare, allo scopo di mantenere in vita nella galassia le ultime vestigia dell'umanità.

688 Attack Sub consente invece agli utenti Amiga di sedere sulla plancia di comando di un modernissimo sommergibile nucleare (c'è la possibilità di sceglierne uno statunitense della Classe Los Angeles o uno sovietico) e di avvalersi della più avanzata tecnologia, per dare la caccia al nemico e vivere, nella fedele riproduzione di affascinanti effetti sonori, la sensazione di trovarsi nelle profondità marine. Al giocatore è concessa la facoltà di distribuire le responsabilità di comando al proprio equipaggio o di condurre la missione accentrando ogni potere decisionale.

Per il C-64 è stato prodotto invece **Ferrari Formula One**, un gioco che permette di simulare la guida della Rossa di Maranello, in 16 diversi circuiti (tra cui Monaco, Monza e Brands Hatch) con la possibilità di scegliere otto tra i migliori piloti di F-1, ognuno dei quali con il proprio personale stile di guida.

Con **Budokan: The Martial Spirit**, il giocatore entra nel suggestivo mondo delle arti marziali e nei segreti della filosofia orientale. Dotato di una grafica realistica e ricca di dettagli, con un'animazione fluida e piacevoli effetti sonori digitalizzati, il programma conduce il "novizio" attraverso una lunga iniziazione, insegnandogli a padroneggiare, sotto la guida di un esperto maestro, le tecniche più micidiali del Karate, del Kendo, del Bo e del Nunchaku.

Electronic Arts
11-49 Station Road, Langley
Berkshire SL3 8XY, England
(Tel. 0044/753/49442)

Classified

Software

Per Amiga scambio programmi di qualsiasi genere. Escluso ogni tipo di lucro. Telefonare, preferibilmente dalle 18 alle 20, al numero 0185/720900 chiedendo di Edoardo – Via Revello, 1-6 – 16036 Recco (GE).

Vendo in blocco 50 dischetti con programmi vari per Amiga a L. 150.000. Telefonare al numero 089/381472 o scrivere a: Carmine Fattorusso – Via A. Volta, 4 – 84098 (SA). Vendo stampante Seikosha SP 1200AI 120 cps NLQ, parallela, a L. 350.000.

Computer e scuola: scambio/offro validissimi programmi didattici ideati e scelti da qualificati insegnanti: programmi per ogni ordine di scuola. Rivolgersi con fiducia, si garantisce serietà e competenza. Andrea Scozzi – Via P. d'Italia, 22 – 73040 Melissano (LE). Tel 0833/921335, ore pomeridiane e serali.

Vendo *Superbase Personal* L. 95.000, *Championship Golf* L. 20.000, *Eco* L. 20.000, *Uninvited* L. 35.000, *Shanghai* L. 15.000. Vendo Sound Buggy per C-64 L. 90.000. Marco Foti – Via Illiria, 18 – 00183 Roma – Tel. 06/777500.

Cerchi programmi per C-128? Se sì, telefona al numero 011/4150675 chiedendo di Alessandro oppure scrivi a Massimiliano Mosca – Via Togliatti, 3 – 10028 Trofarello (TO).

Amiga, scambio programmi. Cerco software per radioamatori, fax, televideo. Dispongo di molto software fatto da me, titoli a richiesta. Walter Radakovic – C.P. 2106/11 – 34100 Trieste.

Cerco/scambio programmi di calcolo strutturale per ing. civile per Amiga. Contatto programmatori. Telefonare ore serali solo dal venerdì alla domenica al numero 099/661022. Annuncio sempre valido.

Scambio software per Amiga nella zona di Pescara. Annuncio sempre valido. Telefonare allo 085/4681740, dalle ore 20 alle 22.30. Chiedere di Alfonso. Escluso giorni festivi.

Scambio qualunque tipo di programma per C-64, dal gioco alle utilities. Invio la mia lista a chiunque ne faccia richiesta. Walter Mughini – Via L. Boccherini, 7 – 50144 Firenze.

Scambio programmi per C-128, sia in modo 128 che in CP/M. Vasta biblioteca software. Annuncio sempre valido. Inviare vostre liste a: Rosario Albanese – C.so Alberto Amedeo, 66 – 90138 Palermo – Tel. 091/327764.

Hai un CBM 64? Sappi che il mio software non conosce limiti e la mia specialità sono i programmi per stampante e gestionali... e se tu volessi creare un videogame... allora rivolgiti a: Luca Lostorto – Via G.B. Vico, 12 – 05100 Terni.

Commodore C-64/128, cedo tutti i migliori programmi (utilities, Geos, grafica, W.P. totocalcio, totip, lotto, games) su disco o cassetta, completi di manuale o istruzioni in italiano. Contattare senza impegno: massima serietà, esperienza, rapidità. Giuliano Cinci – Pian dei Mantellini, 44 – 53100 Siena – Tel. 0577/47054.

Scambio programmi per A/2000. Annuncio sempre valido. Scrivere a: Antonino De Benedetto – Via Svetonio Fabbr. A/11 – 04100 Latina.

Scambio programmi per Amiga video, soprattutto titoli font e animazioni 3D, anche professionali. Scrivere a: Enrico Teperino – Via Iannelli, 45/E – 80131 Napoli.

Cerco disperatamente simulazioni di volo per Amiga 500! Stefano Rossi – Via C. Giovecca, 102 – 44100 Ferrara.

CLASSIFIED DELLA COMMODORE GAZETTE È UN MODO ECONOMICO PER INFORMARE LA PIÙ VASTA UTENZA COMMODORE SUI VOSTRI PRODOTTI O SERVIZI. GLI ANNUNCI NON A SCOPO DI LUCRO, INVIATI DA PRIVATI, VENGONO PUBBLICATI GRATUITAMENTE (COMPILARE L'APPOSITA SCHEDA DI SERVIZIO LETTORI).

Quote: 15.000 lire per linea, minimo 4 linee. Aggiungere 5.000 lire per ogni parola in grassetto o 50.000 lire per l'intero annuncio in grassetto.
Condizioni: pagamento anticipato. Vengono accettati assegni e vaglia postali. Gli assegni devono essere intestati a: IHT GRUPPO EDITORIALE s.r.l.
Forma: gli annunci sono soggetti all'approvazione dell'editore e devono essere scritti a macchina o in modo molto chiaro. Una linea equivale a 40 lettere, spazi tra le parole compresi. Pregasi sottolineare le parole che si intendono scrivere in grassetto.
Informazioni generali: gli inserzionisti devono sempre specificare nome e indirizzo completo. Gli annunci appariranno nel primo numero disponibile dopo il ricevimento.

Inviare il materiale a:
IHT GRUPPO EDITORIALE
UFFICI PUBBLICITARI
VIA MONTE NAPOLEONE, 9
20121 MILANO

Attenzione: Commodore Gazette non si assume responsabilità in caso di reclami di qualunque natura da parte degli inserzionisti e/o dei lettori. Nessuna responsabilità è altresì accettata per errori e/o omissioni di qualsiasi tipo.

Vendo circa 130 dischi di programmi per C-64 + 2 portadischi (90 e 50 pz.) + 2 libri assembler, tutto L. 100.000. Adriano – Tel. 06/6070341.

Scambio o compro programmi di ogni tipo per Amiga e PC MS-DOS, in particolare software musicale (sequencer e campionatori). Inviare lista completa. Vasta disponibilità. Per contatti scrivere a: Giuseppe Ghidella – Via D'Annunzio, 58 – 80053 C/Mare di Stabia (NA).

Scambio programmi per C-64 su disco. Preferibilmente prg. di utilità grafici, musicali, tool, didattici, ecc. Cerco compilatore *Aztec C* della Manx e il *Super C* della Abacus; io ho il computer C-Power della Proline. Max serietà e velocità. Ciro Cané – Via Arcora Comunale, 7 – 80013 Casalnuovo (NA) – Tel. 081/8421498.

Vendo per C-64/128, discoteca 300 giochi su 20 floppy interamente classificata, turbizzata ed etichettata a L. 50.000. Telefonare al numero 091/328755.

Programmi Amiga, C-64, MS-DOS su disco da 3,5" e 5,25". Bollettino software mensile aggiornatissimo. Annuncio sempre valido. Rispondo a tutti. Scrivere a macchina inviando un francobollo a: Maurizio Pastore – Via Vianelli, 10 – 80127 Vomero (NA) – Tel. 081/421146.

Scambio programmi scientifici, grafici, musicali per Amiga 500. Annuncio sempre valido (no lucro!). Scrivere a: Andrea Maggiulli – Via Mascagni, 182 – 00199 Roma. Massima cordialità.

Scambio software Amiga. Max serietà. Non vendo e non compro. Inviare liste a: Sergio Leone – Viale degli Oleandri, 13 – 80131 Napoli.

Scambio programmi di pubblico dominio, esperienze e consigli. Inviare liste e richieste. Rispondo a tutti, massima serietà. Scrivere a: Eugenio Palomba – Via Mura dei Francesi, 195/T – 00043 Ciampino (Roma).

Vendo sistema completo di GEOS con manuali in italiano e in inglese, anche separatamente. Annuncio sempre valido. Telefonare al numero 010/620209.

Vendo giochi e programmi di vario genere per Amiga, causa passaggio a sistema superiore. Dispongo anche di novità. Prezzo da concordare. Per maggiori informazioni o per richiedere la lista, telefonare al numero 0321/27942 (chiedere di Antonio).

Cerco utenti Amiga nella provincia di Latina per scambi software. Giannetti Paolo – Viale della Vittoria, 8 – 04019 Terracina (LT).

Vendo vecchi programmi originali per

C-64/128 e Amiga, dischi vergini, riviste italiane e americane, libri di programmazione, stampante MPS 1200 con carta, a prezzi bassissimi. Solo Roma 06/3765696.

Hardware

Espansione memoria Amiga 1000 esterna Slim Line autoconfigurante da 1 MB L. 400.000. Espansione memoria amiga 500 da 512K L. 150.000, da 1,8 MB L. 600.000. Sconti per quantità. Telefonare ai numeri 06/6115503-7212068.

Affarone! Per cambio sistema, vendo Commodore 64 + disk drive 5,25" + 2 registratori + duplicatore per cassette + joystick indistruttibile + 40 riviste per il C-64 + 400 programmi e giochi tra i più recenti (tra cui *Grand Prix Circuit, Test Drive II, Print Master,* eccetera) + imballi e manuali originali: il tutto in ottime condizioni, all'incredibile prezzo di L. 400.000! Non trattabili. Non lasciatevi scappare questa occasione!!! Max serietà. Filippo Della Bianca – Via V. E. Marzotto, 55 – 30025 Fossalta di Port. (VE) – Tel. 0421/789064.

Vendo espansione di memoria per Amiga 500 nuova da 512K. Richieste L. 160.000. Cedo anche blocco dischetti da 3,5" (100 pezzi) a L. 1.200 cad. – Tel. 0125/712311.

Vendo tastiera, registratore e joystick del C-64, tutto in ottime condizioni: L. 200.000 comprese spese di spedizione e garanzia a 3 mesi. Vendo anche software per Amiga. Vendo espansioni da 1/2 mega a L. 150.000!!! Funzionanti 100% – Telefonare a Paolo, al numero 0746/484988

Vendo interfaccia MIDI per Commodore 64 coi migliori programmi (*Pro 16, MIDI Studio,* eccetera) e manuale. Andrea Laus – Via Pastore, 2 – 20040 Carnate (MI).

Vendo scheda controller Hard Disk A2090A originale Commodore per Amiga 2000, Hard Disk Toshiba A2094 da 40M/28 ms distribuzione Commodore, scheda A2058 espansione RAM fino a 8 megabytes autoconfigurante con 2M montati. Imballi e manuali originali. Luigi Callegari – Via De Gasperi, 47 – 21040 Sumirago (VA) – Tel. 0331/909183 (ore serali).

Vendo scheda Janus XT + drive 5,25", nuovi, con garanzia, per Amiga 2000, scheda 2 MB di espansione originale Commodore. Disponibilità di programmi e manuali. Daniele Zerbini – Tel. 051/6367883.

Vendo Amiga 500 + stampante Commodore MPS 1250 (utilizzabile su Amiga PC, e/o C-64, C-128) + drive esterno (con testina da regolare) + joystick + 20 programmi tra i migliori, a L. 1.120.000. Il tutto è perfetto, imballato ed originale. Vendo anche separatamente. Diego Giorgi - Viale Cortemaggiore, 12 – 93012 Gela (CL) – Tel. 0933/938404.

Cerco urgentemente drive x CBM 64 in buono stato a non più di L. 180.000. Annuncio sempre valido. Gabriele Sernia – Via Marconi, 17 – 80046 S. Giorgio a Cremano (NA) – Tel. 081/270976 (dalle 10 alle 13).

Acquisto monitor 1084 della Commodore ad un prezzo di L. 350.000 trattabili. Telefonare al numero 0584/928807.

Vendo espansione per Amiga 500 e 200 dischi 3,5" bulk Sony. Scrivere o telefonare a: Silvano Bompieri – S.da dei Colli, 60 – 46040 Monzambano (MN) – Tel. 0376/800772.

Cerco buona stampante usata, compatibile con A500. Roberto Rota – Via Portichetto, 9 – 24052 Azzano S. P. (BG).

Vendo espansione da 512K interna per A500, nuova, a L. 160.000. Cedo anche mouse 1351 per C-64, usato a L. 40.000 (molto SW in omaggio). Massima serietà. Gianni Cottogni – Via Strambino, 23 – 10010 Carrone (TO) – Tel. 0125/712311 (ore serali).

Vendo scheda Janus XT (rende Amiga compatibile IBM) ancora da acquistare, perché oggetto indesiderato facente parte di offerta Commodore. Garanzia Commodore. Telefonare al numero 0544/31249 (Mario) o al numero 02/2482595 (Gianni).

Vendo MIDI professionale con sequencer per Amiga (la confezione non è mai stata aperta!!!) + in regalo 20 giochi o utility a vostra scelta!!! Il tutto a L. 100.000!!! Telefonare al numero 0523/65756.

Cerco espansione di memoria 1750 per C-128 (512K). Scambio inoltre programmi per C-64 (giochi, grafica, musica), C-128 (possiedo il *Basic 8.0* e il CP/M (Turbo Pascal). Telefonare a Roberto Caporale: 071/81157 (ore pasti).

Vendo Commodore Pet 3032 + tastiera + monitor FV + registratore + bifloppy 3040 + stampante 3022, a L. 890.000. Regalo software professionale e istruzioni travaso programmi dal C-64. Telefonare al numero 091/328755.

Cedo hardware per qualsiasi esigenza (per C-64, C-128 e Amiga): drive, espansioni, dischetti eccetera. Telefonare al numero 080/686059 (ore serali).

Vendo stampante MPS 1500 acquistata il 2/8/89, ottimo stato, con manuale italiano, imballaggi originali, garanzia triennale, inchiostro colorato, interfaccia Centronics e 2000 fogli, a L. 450.000 (trattabili). Rivolgersi a: Mario Corsolini – Via Bicocchi, 3 – 58022 Follonica (GR) – Tel. 0566/55024.

Attenzione!!! Vendo Amiga 500 (nuovo) + disk drive slim + espansione RAM interna da 512K (con clock) + 2 joystick + elegante mobile portacomputer e disk drive, a L. 1.100.000. Telefonare al numero 0771/741263 (Luigi).

Varie

Vendo per TI/99, cartuccia SSS "Miny Memory" + due libri sulla programmazione in linguaggio macchina del TMS 9900 + cartuccia SSS "Parsec". Prezzo da concordarsi. Vendo inoltre pocket computer Sharp PC1401, programmabile in Basic e l. m.. Telefonare a Walter al numero 0171/62767.

Scambio la mia collezione di francobolli della Repubblica Italiana, composta di 850 francobolli nuovi e usati e di 4 album GBE Milord per un valore complessivo di L. 1.350.000 (+ altri regali filatelici) per un Amiga 500. Per chiarimenti scrivere o telefonare a Bruno Gallo Giuseppe – Via Como, 79 – 92015 Raffadali (AG) – Tel. 0922/471018 (ora pranzo).

Cerco materiale riguardante pacchetti per C-64 e/o C-128 applicati alla musica... batterie elettroniche, sintetizzatori, campionatori, eccetera. Contattatemi. Matteo Mairate – Via Fiume, 96 – 10088 Volpianto (TO).

Vendo riviste: *MC Microcomputer* dal n. 43 al n. 53 (escluso n. 45); *SuperVic e C-64* dal n. 2 al n. 6; *SuperCommodore* dal n. 1 al n. 7; *Noi 64 e 128* i primi numeri. Tutte tranne *MC*, con cassette programmi. Prezzi di copertina. Antonio Di Palma – Via Petrosini, 10 – 84014 Nocera Inferiore (SA) – Tel. 081/923810.

Sono disponibili le seguenti traduzioni in italiano: *Geos 128* (L. 70.000); *Deskpack Plus 128* (L. 50.000); *Font Master 128* (L. 60.000); *Spell Master 128* (L. 30.000) più spese postali. Telefonare al numero 0985/85055, dopo ore 21.

Amiga, tutto il suo mondo, con proposte ecc. Anche scambi con programmi, giochi e tutto ciò che lo riguarda, valido anche per il C-64 e lo ZX Spectrum 48K. Scrivere a: Renzo Reggio – Via Roma, 57 – 00032 Carpineto Romano (ROMA) – Tel. 06/9797134 (sera).

Vendo *Rom Kernel Reference Manual: Libraries and Devices* in più che ottimo stato a L. 40.000. Inoltre scambio sorgenti assembler per Amiga. Pietro Montelatici – Via Baracca 9/11 – 16035 Rapallo (GE) – Tel. 0185/60829.

Cerco informazioni per iscrizioni Amiga Club con biblioteca software aggiornata. Vito Attolico – Viale S. Antonio, 2 – 08015 Macomer (NU).

Vendo arretrati di *Commodore Gazette* e *AmigaWorld* a metà prezzo. Vendo sistema Geos completo, con manuali per Commodore 64. Luciano Flennio – Via Pian di Forno 22/9 – 16153 Genova Sestri Ponente – Tel. 010/620209.

Vendo a metà prezzo di copertina e assolutamente nuovi: *Amiga Handbook* (Jackson L. 16.000), *Il manuale dell'Amiga*

DOS (I.H.T. L. 30.000), *L'Amiga* (I.H.T. L. 30.000). A chi li compra assieme, regalo *Amiga Basic* originale Commodore in italiano. Salvatore Ripoli – Via Giovanni Ninni, 14 – 80135 Napoli – Tel. 081/5519262.

COMMODORE COMPUTER SERVICE – Centro riparazioni e servizi di assistenza tecnica per computer, periferiche e accessori COMMODORE. Effettuiamo riparazioni in 48 h. con preventivo gratuito e possibilità di ritiro/consegna a domicilio. Per qualsiasi informazione, telefonare a: Antonio Piscopo – Via S. Lucia Filippini, 49 – 80142 Napoli – Tel. 081/284246 ore ufficio.

Utente Amiga cerca appassionati adventure per aiuti e soluzioni. Vasta biblioteca con novità. Contattare: Vasco Leporini – Via del Popolo, 37 – 51019 Ponte Buggianese (PT) – Tel. 0572/635050 ore 19/21.

Commodore Club

Alessi Software invita tutti gli amighi a far parte dell'Amiga Club per scambio esperienze e programmi di PD: no lucro, astenersi pirati. Scrivere a: Fabio Alessi – Via Milano, 5 – 22077 Olgiate Comasco (CO) – Tel. 031/944128.

LEOSOFT CLUB Amiga – IBM – C-64, a disposizione tutte le novità. Si scambia inoltre soft per Amiga e C-64, per soft IBM e compatibili. Mandateci le vostre liste MS-DOS, nessuno scopo di lucro. Leosoft Club – Via S. Nicastro, 4 – 91026 Mazara del Vallo (TP) – Tel. 0923/945623.

Il Software Computer Club – P.O. Box n. 39 – 13060 Valdengo (VC) scambia programmi per Amiga 500/1000/2000 e per IBM XT/AT (possediamo Amiga 2000 con scheda 2286). Se abiti in zona, telefonaci: 015/680743 (Luca), 015/680468 (Doriano). Saremo lieti di venirvi a trovare a casa.

Cerco utenti Amiga per formare un club. Già disponibili più di 2000 programmi. Per informazioni, scrivere a: Alessandro Sancolodi – Viale Venezia, 40 – 36061 Bassano del Grappa (VI) – Tel. 0424/31878.

Amiga Club cerca nuovi soci: scambia programmi per Amiga, disponibili ultime novità. Massima serietà. Tel. 0185/51320.

Iscrivetevi al primo club che invia ogni mese a tutti i soci una rivista con una scelta dei migliori videogiochi e novità per Commodore 64! Per informazioni, contattateci subito... scrivendo a Rocco Gregorio – Via Appia, 65 – 85050 Baragiano Sc. (PZ) o telefonando al numero 0971/993258 (Piero).

Il Commodore Club Catania cerca nuovi soci in tutta Italia (isole comprese). Il nostro club vi propone iniziative interessanti: software, hardware, bollettino mensile, novità. Telefonate o scrivete: Santi De Luca – P.zza dei Martiri, 3 – 95131 Catania – Tel. 095/531198.

Gruppo utenti Amiga Finale Ligure contatterebbe possessori Amiga, interessati a partecipare attivamente all'attività 1990. Gruppo Utenti Amiga – P.zza Santa Caterina – 17024 Finale Ligure (SV).

Dany is proud to present a new club. If you wanna get in touch with me, you can't but call this hot line: 0825/71710. A wide range of new hot stuffs is waiting for you. Also manuals, books, tricks, intro, BBS numbers. Wake up and call me! ■

INDICE DEGLI INSERZIONISTI

Direzione vendite spazi pubblicitari:

IHT Gruppo Editoriale – Commodore Gazette
Agente Pubblicitario: Aldo Pagano Pagano
Via Monte Napoleone, 9 – 20121 Milano
Tel. 02/794181 - 799492 - 792612 - 794122
Telex 334261 IHT I – Telefax 02/784021

Questo indice è da considerarsi come un servizio addizionale. L'Editore non si assume alcuna responsabilità per eventuali errori e/o omissioni.

Indirizzare eventuali lamentele riguardanti gli inserzionisti a:

Commodore Gazette
Uffici Pubblicitari
Relazioni Inserzionisti
Via Monte Napoleone, 9 – 20121 Milano

Nessuna responsabilità viene altresì assunta dalla Commodore Gazette per eventuali problemi di qualsiasi natura con gli inserzionisti. La responsabilità di quanto pubblicato negli spazi pubblicitari è esclusivamente del committente.

COME DIGITARE I LISTATI DI COMMODORE GAZETTE

I listati per C-64/128 contengono una particolare simbologia. Tutti i caratteri grafici e quelli di controllo sono stati tradotti in combinazioni di tasti facilmente comprensibili. Sono le istruzioni tra parentesi graffe. Per esempio, {SHFT L} indica che si deve tener premuto il tasto shift e premere una volta il tasto L. Ovviamente, non bisogna digitare le parentesi; quello che apparirà sullo schermo saranno simboli grafici. Altri esempi:

{20 SPAZI} premere la barra spaziatrice 20 volte.

{SHFT CLR} tenere premuto il tasto shift e premere una volta il tasto clr-home.

{2 CRSR ↓} premere cursore-giù due volte.

{CTRL 1} tenere premuto il tasto control e premere il tasto 1.

{COMD T} tenere premuto il tasto col logo Commodore e premere T.

{CRSR ←} premere cursore-sinistra una sola volta.

{SHFT A} tenere premuto il tasto shift e premere il tasto A.

Gli altri tasti che non danno origine a caratteri particolari (come ↑, £, @) sono invece presentati normalmente.

Manoscritti: le collaborazioni dei lettori – manoscritti, disegni e/o fotografie – sono benvenute e verranno valutate in vista di una possibile pubblicazione. Commodore Gazette non si assume comunque responsabilità per perdite o danni al materiale. Si prega di allegare una busta affrancata e indirizzata per ogni articolo. Il pagamento per materiale non richiesto viene effettuato solo in seguito all'accettazione da parte della redazione. I contributi editoriali (di qualunque forma) non si restituiscono. Tutta la corrispondenza editoriale, richieste di annunci, problemi di sottoscrizione abbonamenti, di diffusione e con gli inserzionisti, deve essere indirizzata a: Commodore Gazette – Uffici Editoriali – Via Monte Napoleone, 9 – 20121 Milano. **Commodore Gazette** è un periodico indipendente non connesso in alcun modo con la Commodore Business Machines e con tutte le sue sussidiarie e affiliate, compresa la Commodore Italiana S.p.A. Commodore Gazette viene pubblicata dalla IHT Gruppo Editoriale, Via Monte Napoleone 9, 20121 Milano.

IL PROSSIMO NUMERO SARÀ IN EDICOLA ALLA FINE DI MAGGIO

SERVIZIO LETTORI

Questa scheda è valida fino al 31 maggio 1990

A. Come giudica questo numero di Commodore Gazette?

- ☐ 1. Ottimo
- ☐ 2. Molto buono
- ☐ 3. Buono
- ☐ 4. Discreto
- ☐ 5. Sufficiente
- ☐ 6. Mediocre
- ☐ 7. Insufficiente

B. Quale(i) articolo(i) di questo numero ha apprezzato maggiormente?__

C. Quale(i) articolo(i) di questo numero giudica peggiore(i)?__

D. Quali argomenti dovrebbero essere trattati nei prossimi numeri di Commodore Gazette?__

E. Con quale aggettivo descriverebbe Commodore Gazette?__

F. Quante persone leggono la sua copia di Commodore Gazette?

- ☐ 1. Uno
- ☐ 2. Due
- ☐ 3. Tre
- ☐ 4. Quattro o più

G. Ha dei suggerimenti?__

H. Quale(i) computer utilizza?

- ☐ 1. C-64
- ☐ 2. C-128
- ☐ 3. C-128D
- ☐ 4. Amiga 500
- ☐ 5. Amiga 1000
- ☐ 6. Amiga 2000
- ☐ 7. Altro (specificare)__

I. Quale(i) computer intende acquistare nel futuro?

- ☐ 1. C-64
- ☐ 2. C-128D
- ☐ 3. Amiga 500
- ☐ 4. Amiga 2000
- ☐ 5. Altro (specificare)__

L. È un acquirente dei libri della IHT? Se sì come li giudica?__

M. Quali altre riviste (sia d'informatica che non) legge abitualmente?__

N. Indichi quali sono i suoi maggiori interessi

- ☐ 1. Videoregistrazione
- ☐ 2. Hi-Fi
- ☐ 3. Strumenti musicali
- ☐ 4. Fotografia
- ☐ 5. Automobili
- ☐ 6. Sport
- ☐ 7. Viaggi

O. Quali periferiche intende acquistare nei prossimi sei mesi?__

P. Quanto intende spendere in software e hardware nei prossimi sei mesi?__

Nome __
Cognome __
Indirizzo __
Città __
Prov. __ **C.a.p.** __ **Età** __
Professione __

COMMODORE GAZETTE **aprile 1990**

SCHEDA ORDINAZIONE LIBRI

Con il presente tagliando desidero ordinare i(l) seguenti(e) libri(o):

Collana Informatica

- ☐ L'Amiga (Michael Boom) .... L. 60.000
- ☐ Il Manuale dell'AmigaDOS (Commodore-Amiga) .... L. 60.000
- ☐ Programmare l'Amiga Vol. I (Eugene P. Mortimore) .... uscita: aprile '90
- ☐ Programmare l'Amiga Vol. II (Eugene P. Mortimore) .... L. 70.000
- ☐ Guida ufficiale alla programmazione di GEOS (Berkeley Softworks) .... L. 64.000
- ☐ Flight Simulator Co-pilot (Charles Gulick) .... L. 30.000
- ☐ Volare con Flight Simulator (Charles Gulick) .... L. 45.000

Collana Cinema

- ☐ Le mille luci di Hollywood (David Chell) .... L. 42.000

Collana Tempus

- ☐ Inventori del nostro tempo (Kenneth A. Brown) .... L. 42.000
- ☐ Computer in guerra: funzioneranno? (David Bellin e Gary Chapman) .... L. 39.900
- ☐ La sfida della crescita (G. Ray Funkhouser e Robert R. Rothberg) .... L. 39.900
- ☐ La Macchina e la Mente (George Johnson) .... L. 42.000

Pagherò in contrassegno al postino la somma di L.......... + spese postali (L. 6.000 per volume)

Nome e cognome __
Indirizzo __
Città __
Prov. __ **C.a.p.** __ **Tel.** __
Firma __

COMMODORE GAZETTE **aprile 1990**

☐ Desidero inserire gratuitamente un mio annuncio nella rubrica CLASSIFIED (solo per i privati e per gli annunci non a scopo di lucro).

☐ Desidero inserire un mio annuncio nella rubrica CLASSIFIED. Allego assegno bancario o circolare o postale oppure fotocopia della ricevuta del vaglia postale per un totale di L. __________________ . Il mio codice fiscale o partita IVA (per le aziende) è il seguente: ______________________.

Attenzione: perché un annuncio venga accettato è necessario che sia stato compilato anche il questionario presente sull'altro lato di questo tagliando. Non si accettano fotocopie.

TESTO: ____

Inserire all'interno di una busta affrancata e spedire a:

Commodore Gazette
Servizio Lettori
Via Monte Napoleone, 9
20121 Milano

- - - - - ✂ -

Inserire all'interno di una busta affrancata e spedire a:

IHT Gruppo Editoriale
Divisione Libri
Via Monte Napoleone, 9
20121 Milano

UN COMPUTER, UN LIBRO...
...LE ALI

FLIGHT SIMULATOR CO-PILOT

Questo volume è un vero "istruttore di volo" per coloro che possiedono il programma Flight Simulator della Microsoft per i PC IBM e compatibili, oppure Flight Simulator II della SubLogic per Commodore 64, Apple II, Atari 800 XL e XE.

Non è un semplice compendio di comandi da ricordare a memoria, ma un brillante compagno d'avventura, scritto con estro e ironia. Un libro da sfogliare con il computer acceso, per imparare passo dopo passo i principi del volo "rettilineo e livellato", le procedure di decollo e di atterraggio, del volo strumentale e notturno, e del volo acrobatico.

152 pagine, L. 30.000

VOLARE CON FLIGHT SIMULATOR

Questo volume si rivolge a tutti gli utenti che possiedono la versione di Flight Simulator per l'Amiga, l'Atari ST o l'Apple Macintosh.

Con uno stile spigliato, ma con grande attenzione a ogni dettaglio, l'autore conduce l'allievo pilota attraverso le varie fasi di un corso di volo che si concluderà con un primo approccio al volo acrobatico. Il testo è completato da dettagliate cartine di rotta, schemi che illustrano l'uso dei comandi, parametri di volo ideali e procedure utili nelle più svariate situazioni di volo.

232 pagine, L. 45.000

I libri IHT sono disponibili nelle migliori librerie e computer shop. Per ordini diretti servirsi dell'apposito modulo pubblicato a pagina 127
IHT Gruppo Editoriale - Via Monte Napoleone, 9 - 20121 Milano - Tel. 02/794181-794122 - Fax 784021 - Telex 334261 IHT I
Distribuzione: RCS Rizzoli Libri - Via Scarsellini, 17 - 20161 Milano - Tel. 02/64068508